Ben-Gurion University of the Negev
Department of Mechanical Engineering

Stabilization of an Underactuated Robot on a Ball

By: Alon Ohev - Zion
Supervisor: Dr. Amir Shapiro

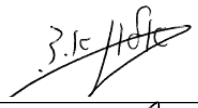
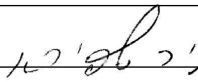
Submitted in Partial Fulfilment for the
M.Sc. Degree in Mechanical Engineering

Ben-Gurion University of the Negev
Department of Mechanical Engineering

Stabilization of an Underactuated Robot on a Ball

By: Alon Ohev - Zion
Supervisor: Dr. Amir Shapiro

December 2009

Alon Ohev - Zion:		Date: December 16, 2009
Dr. Amir Shapiro:		Date: December 16, 2009
Prof. Alexander Yakhot :		Date:

Submitted in Partial Fulfilment for the
M.Sc. Degree in Mechanical Engineering

Abstract

In this work we investigate and derive a nonlinear control law for a spatial model of the underactuated Ballbot system. The spatial Ballbot system is an extended 3D model to the 2D model of a unicycle. The main classes of inverted pendulums are reviewed, and their principle differences are considered. The holonomic nature of the motion of the ball rolling on a plane is discussed, and a new innovative approach considering the motion of the center of the ball as holonomic is described. A spatial 3D kinematic and dynamic model are derived, and a control law for the stabilization and position of the robot are constructed. Partial feedback linearization for an underactuated systems, is used to derive the stabilization controller, and a simple P.D. controller is used for the position control. The stability nature of each controller is investigated, and the overall system is proven to be locally stable. Computer simulation are presented, and the experiments are preformed using the LEGO NXT Mind-Storm kit.

Acknowledgments

I wish to express my gratitude to my supervisor, Dr. Amir Shapiro. who was abundantly helpful and offered invaluable assistance and support, and offered his precious laden time.

I would like to specially thank Prof. Emeritus Ben-Zion Sandler, who was the proponent of the project, guided me through my first steps in research and found the bases to my work.

I gratitude Dr. Yizhar Or, for his precise observations and clarifying comments, witch contribute to the accuracy of this work.
Special thanks to me friend and office-mate Mr. Alon Capua, who's advices, reasoning and laughs were the bases for the everyday life.

I am grateful to the "Yaacov Ben-Izhak Hcohen" scholarship fund, for its great support and assistance.

Many thanks to the Mechanical Engineering department's technicians crew, for their help and experienced advice.

Last but not least I wish to express my love and gratitude, to my beloved family, for their understanding, endless love, through the duration of my studies.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The aim of this work, is to investigate and derive a nonlinear control law for a spatial inverted pendulum. The main motivation, is to improve the modelling and control of a "Ballbot" like system, *i.e.* a mobile, human hight, narrow and single spherical wheeled robotic platform. This kind of robot can be very useful in handling chores, while being assimilated among people in work places and at home.

Next are main needed terms. *DoF (Degrees of Freedom),* are the minimum number of parameters needed, in order to fully define the system configuration. It is now possible to describe the set of all possible configurations of the DoF , as the *Configuration-Space*. The configuration space is a part of the *State - Space*, which is defined by all the configurations of the DoF and theirs time derivatives. The state space is used to describe the system's dynamic behavior. An Equilibrium Point, is a point in the state space which describe the system configuration so it is kept static. We denote two types of equilibrium points, Stable and Unstable. A point is called a *Stable Equilibrium Point*, if after a slight deviation of the system off that point, the system will tend to return to it. An *Unstable Equilibrium Point*, is a point where after a slight deviation of the system off it, the system will tend to draw away from it. Considering the Hamilton principle [**14**], it is understandable that the stable equilibrium point is a local minimum, and

the unstable equilibrium point is a local maximum of the mechanical system's (without the controller) energy function.

Our subjected Ballbot system, can be modeled as a Wheeled based inverted pendulum. A vertical pendulum has two equilibrium points. An upper unstable one, where the pendulum's center of mass is positioned on the vertical line above the pendulum's pivot, and a lower stable one, where the pendulum's center of mass is positioned on the vertical line beneath the pendulum pivot. In order to grasp the concept, four different types of planar mathematical inverted pendulums[1] are introduced Figure 1.

A *Fixed Pivot inverted pendulum*, Figure 1.1a, is the most intuitive in[**30**]verted pendulum. In order to keep this pendulum in it's upper unstable equilibrium point, a control torque is applied about it's pivot.

A *Cart inverted pendulum*, Figure 1.1b, consist of a free rotating pivot mounted on a cart. The idea is to bring the pendulum's pivot underneath the pendulum's center of mass. By doing it, the pendulum is positioned at it's upper unstable equilibrium point. In order to achieve this, an horizontal controlled force is applied on the cart. This system is underactuated, due to the free rotating pivot. That character cause one of the main difficulties in stabilizing this system.

A *Wheeled inverted pendulum*, Figure 1.1d, is in a sense a combination of the first two. The pendulum's pivot positioned at the center of a wheel, and the control torque is applied between the pendulum and the

[1] a pendulum consist of a massless rode and a point mass.

wheel. The applied torque in fact preform two functions. On the one hand, it pushes the pendulum upwards to it's equilibrium point - like the fixed pivot pendulum. On the other hand, it rolls the wheel in the other direction, and by that position the pendulum's pivot underneath it's center of mass - like the cart inverted pendulum. This system is as well underactuated, due to the coupling between the pendulum's angle and the position of it's pivot - the wheel center.

A *Parametric pendulum*, Figure 1.1c, has a slightly different concept than the previewed pendulums. Parametric pendulum consist of a torque free pivot, which is rapidly oscillating along the desired alinement direction. It will be shown here, that these vibration causes the pendulums to aline with the line of action of the vibrations. This phenomenon can be used in order to stabilize an inverted pendulum at it's upper unstable equilibrium point.

In Chapter 2, literature survey is presented. Reviewing existing ballbot like platform and the control of an underactuated systems. In Chapter 3, Kinematics and dynamics of a three dimensional (3D) spatial wheeled inverted pendulum is presented. In Chapter 4 a non-linear control law is derived, considering the full 3D non-linear system, which is one of the main contributions of this work. Chapter 5 presents simulations and experimental results of the system stability, by demonstrating the controller's capabilities of handling initial conditions out of the linearity boundary. Chapter 6 presents conclusions and future work.

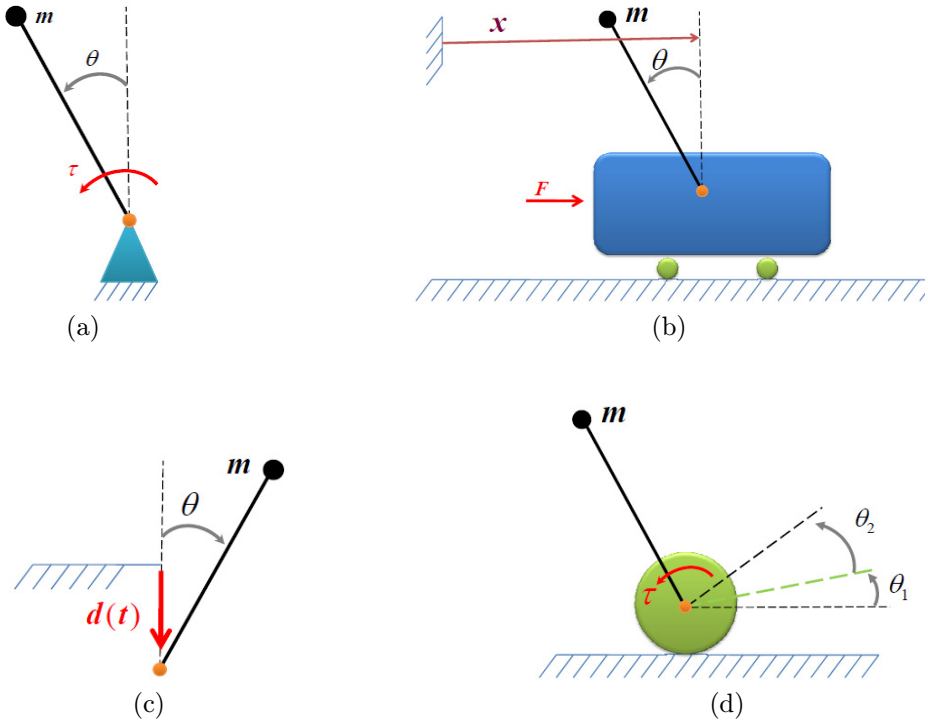


Figure 1.1: Four different concepts to balance an inverted Pendulum. (a) Fixed pivot Inverted Pendulum. (b) Cart Inverted Pendulum. (c) Parametric Pendulum Inverted - Rapidly oscillating pivot. (d) Wheeled Inverted Pendulum.

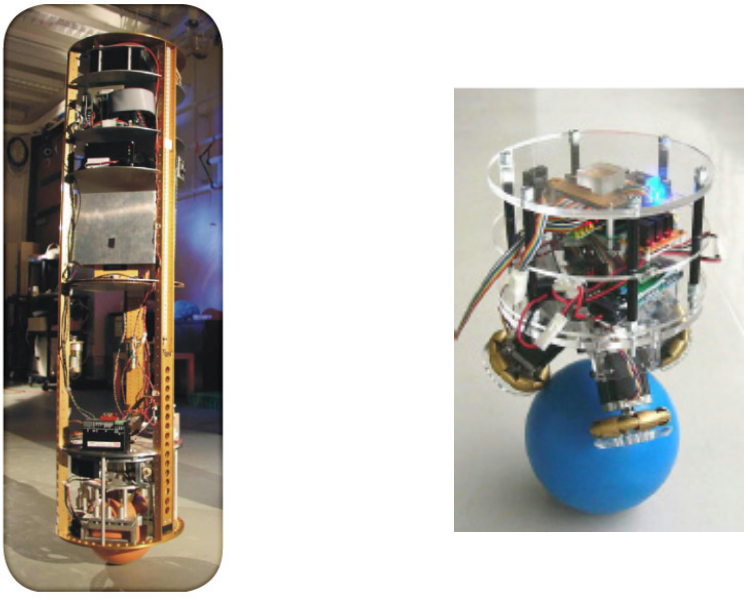
CHAPTER 2

Literature Survey

The idea of single-wheeled robot, like many of the technological development, originally emerge in the science fiction stories, like Cluster by Piers Anthony. The first attempts to build this kind of robot, considered a rugby shaped ball as a wheel [27]. The robot's wheel architecture consist of an axel alone the main axis of the rugby shaped wheel, and a perpendicular cylinder joint. Blanc in the forward/backward direction was achieved by the solving the planner inverted pendulum - unicycle control problem, while the sideways balance was achieved by tilting the robots body about the perpendicular joint. The increasing radius of the wheel in the sideways directions, simplified the control law and the need for a spherical actuator.

Two of the known autonomously balanced systems are the "Segway" [2, 28] and the balancing wheel chair [40, 1]. This systems are robust and capable of handling non-horizontal and un-smooth terrain. The control and maneuvering of an inverted pendulum is widely studied in [20]. A wide variety of mobile robots for a range of application based on the Segway platform exist, as in [28, 24, 42, 9] .

The first real spherical wheel robot was established by T.B. Lauwers, G.A. Kantor and R.L. Hollis [3, 37, 38, 39] from CMU. They built a human hight robotic platform that maneuver itself by rolling



(a) (b)

Figure 2.1: Two Ballbot systems. (a) Ballbot of C.M.U. [32]. (b) Omni-wheeled Ballbot [23].

on a ball. It is obvious that this system is not statically stable and need a control system in order to obtain dynamic stability. In order to achieve that, they simplified the problem, by modeling the robot as two decoupled 2D linear unicycles. This separation can be seen in the architecture of the inverse mouse-ball drive train, and in the separate controllers. The applied robot architecture has the advantages of human hight, holonomic mobility[1] and a slim structure. This advantages are important while considering the robot's assimilation among humans. On the other hand, it has two main disadvantages. First, the controller is limited and bounded, because it is built considering a first degree - linear approximation model of the dynamic system, and assumes decoupling of the special angels. Second, the inverse mouse-ball drive mechanism cause a redundant friction, and is pivoting incapable.

[1]The ability to preform movement in any direction in the workspace

Considering a solution for the redundant friction and pivoting incapability, rise the need for a spherical actuator, that is more efficient than the inverse mouse-ball drive train. A good spherical actuator, propel by an omni-directional wheels, was suggested by J.B. HOLLAND et al.[**16**]. The use of omni-wheels provide virtually friction-free motion parallel to each omni-wheel rotation axis, creating the potential for unconstrained angular motion. Since the omni-wheels contacts the ball directly, there are no joints or links interfering with its motion, allowing full 360° rotation in all axes.

In 2008 Masaaki KumagBai and Takaya Ochiai published and presented a Ballbot driven by an omni-directional spherical actuated [**23**], Figure 2.1b. The control law of this ballbot is based on double 2D linear approximation of the dynamic model. The controller generate two separate control signals - one in each direction - each based on a Linear Quadratic Regulator control. The controller consider the orientation and position of the robot. This two signals is later transformed to the omni wheels actuator via kinematical calculation.

The model concept of this kind of system is referred as an inverted pendulum. As seen in Figure 1, we define four different concepts of planer inverted pendulum configurations: Fixed pivot, Cart , Parametric and Wheeled Inverted Pendulums. In 1995 Qifeng [**33**] showed that the minimum horizontal distance a cart inverted pendulum must travel in order to lift itself to its upper unstable equilibrium point, while it is not allowed to swing through its downwards equilibrium point, is equal to the horizontal projection of the *effective pendulum*. The effective pendulum is a known transformation made in order to represent

the pendulum as it consist of weightless rod and a point mass [**33**], by that simplify the analytic exploration of the system. In 2006, A.M. Formal'skii [**12**] showed that under linear controller limitations, a wheeled inverted pendulum has a bigger controllability domain, of initial conditions, then a Fixed pivot one. Therefore a wheeled inverted pendulum is more easily stabilised under same initial - conditions using a linear controller. Considering Qifeng [**33**] and Formal'skii [**12**], we anticipate that the wheeled inverted pendulum has the advantages of minimal horizontal base travel - in order to bring the pendulum to the upper unstable equilibrium point, and better initial conditions handling than the Fixed pivot or cart inverted pendulums.

One of the difficulties in the kinematics modeling of the spacial ballbot system, is to describe the roll of the ball over the horizontal plane, due to the non-holonomic nature of this motion. As detailed later in Section 3.1, Bloch and borisov [**7, 8**] shown that it is possible to reorient a ball as desired, by rolling it along a close path on the horizontal plane. That means that there is not a direct relation between the position of the ball's center and its orientation *i.e.* non-holonomic. Simplified model assuming slip less roll with no rotation about the vertical -pivoting, was developed by C.Camicia, F.Conticelli and A.Bicchi for the "Spherical" - a robot that maneuver by driving inside a sphere [**11, 6**]. As known, the slip less roll constraint is a simplified approximation of the rolling kinematics, and a minor slip always exist. Modeling the slip friction influence, as published by [**16, 22, 34**], in the overall kinematic model is one of the long term goals.

The ballbot is an *Underactuated mechanical system, i.e.* it has less actuators then DoF . Two classes of an underactuated mechanical systems, as defined by M.W.Spong [35], for systems with n DoF are *upper* and *lower*. An *Upper* actuated system is where the first m DoF are actuated, and *Lower* actuated system - where the last $(n-m)$ DoF are actuated. Obviously, Ballbot is a lower actuates system, as we consider the ball's DoF as first. Approaches in handling this class of system described in [26, 31, 29, 36], and are considered in this work.

A different, and maybe less intuitive method to stabilize an inverted pendulum, is called *Parametric pendulum*, Figure 1.1c. This type of pendulum was described by Panovko and Govnov in 1967 [18]. Parametric pendulum consist of a torque free pivot, which is rapidly oscillating along the desired alinement direction. Because of the rapidly change in the pendulum's pivot position, the pendulum situated in rapidly cyclic acceleration field. Kapitza and Butikov [18, 10] described and analysed this phenomena by considering a non-inertial positioned at the pendulums mass. Due to the acceleration of this frame, the *D'Alembert Principle* of "apparent force" is considered, in order to explain the converges of the pendulum position. Another analysis, considering an inertial frame was published by Acheson [4, 5] in 1993-1995. All of the analyses, manipulate the equations of motion in to a non-dimensional form. A linearization about equilibrium point produce a differential equation, known as the *Mathieu equition*. This equation is known to describe the dynamic stability, generated by oscillations, as in this case [13].

CHAPTER 3

Kinematics and Dynamics Of Ballbot

This chapter present the kinematics and dynamics equations of the full 3D spatial Ballbot system. The kinematic equations are used to describe the relation between the positions, velocities and accelerations of each component constructing the system. All with respect to the system's DoF . Later, based on the system's kinematics, the dynamic equations are derived, using the Lagrangian and the Euler-Lagrange equation, and formulated into matrix form. The dynamic equation are the reduced form of Newton's equations of motion, represented by their dependency at the chosen DoF . The derivation of the matric form equations of motion is a closed form [26].

3.1. Kinematics

Let us consider a Ballbot, build out of a rigid main body, rolls on a rigid ball on the horizontal $x - y$ plane, relative to the inertial coordinate frame $F_0 = [x_0, y_0, z_0]^T$ with origin O_0. We shall define a frame parallel to the inertial frame F_0, positioned at the ball's center of mass (cm_b) as $F_b = [x_b, y_b, z_b]^T$, with origin O_b. Define another frame, fixed to the robot‘s main body, placed at its center of mass (cm_r) as $F_r = [x_r, y_r, z_r]^T$, with origin O_r.

In order to describe the translation and rotation of coordinate frame F_i relative to coordinate frame F_j, we shell use the *Homogeneous Transformation Matrix*: ${}^jA_i$. e.g. In order to represent the vector r_i whose values are given relatively to translated and rotated coordinate frame F_i, in respect to coordinate frame F_j, we preform:

$$\begin{bmatrix} r_j \\ 1 \end{bmatrix} = \begin{bmatrix} {}^jA_i \end{bmatrix} \begin{bmatrix} r_i \\ 1 \end{bmatrix}$$

and:

$$[{}^jA_i] = \begin{bmatrix} [{}^jR_i] & {}^j\!b \\ 0_{1\times 3} & 1 \end{bmatrix}$$

Where $[{}^jR_i] \in SO3$, is the rotation matrix, describes the rotation of F_i in respect to F_j, and ${}^j\!b$ is the translation of the origin O_i in respect to the origin O_j.

The DoF are chosen in a different way for the ball and for the body. In order to describe a ball rolling on a plane we need five DoF , (x, y) coordinates of its center, and three angles to describe its orientation. It is widely known, that the rollover of a ball on a plane is a non-holonomic motion. That is because the orientation of the ball's fixed frame after sequence of rotations is directly affected by the order and magnitude of each rotation. As detailed in [**7**, **8**, **17**], it is possible to reorient a ball as desired, by rolling it along a close path on the horizontal plane. Meaning that there is not a direct relation between the position of the ball's center and its orientation *i.e.* non-holonomic. As we would like to describe the relation between the ball's rotation to

its location, we have to use the non-holonomic constraint:

$$\begin{cases} \dot{x}(t) - r\omega_{y_0}(t) = 0 \\ \dot{y}(t) + r\omega_{x_0}(t) = 0 \end{cases}, \tag{3.1}$$

where $x(t), y(t)$ are the position of the ball's center, and $\omega_{x_0}(t), \omega_{y_0}(t)$ are the angular velocities of the ball in the inertial frame. This constraint adds two equation to compute and transform the system to be not integrable. In order to overcome this obstacle, we note that the ball's orientation is not important to us, only the position of its center is relevant. Further more the ball holds special spatial symmetry, as its inertia matrix is invariant under frame rotations. By comprehend the last two notes, we see that the elements of $\vec{\omega}_{b_0}$, that is the ball's angular velocity vector given relative to the inertial frame, are independent while regarding the position of the ball's center. In that case,

$$\int_t \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} dt = \begin{bmatrix} \theta_1(t) \\ \theta_2(t) \\ \theta_3(t) \end{bmatrix} + \vec{c}.$$

Same as $(\omega_x, \omega_y, \omega_z)$, $(\theta_1(t), \theta_2(t), \theta_3(t))$ are independent, and $\vec{c}$ is the integration constant vector, defined by the initial conditions. Although these $(\theta_1(t), \theta_2(t), \theta_3(t))$ are not the rotation angles used to describe the ball's orientation, we will use them as the ball's DoF . One can think of this DoF as the rotation preformed by the CMU Ballbot actuator rolls, as can be seen in Figure 3.1. It is now possible to produce an

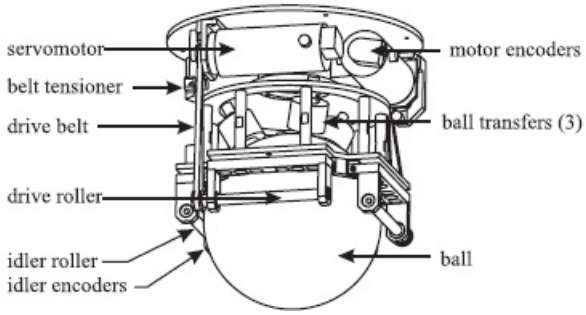


Figure 3.1: Inverse mouse drive trine, used by CMU Ballbot [21]

holonomic non-slip constraint between the ball's DoF :

$$\begin{cases} x(t) - r\theta_2(t) = 0 \\ y(t) + r\theta_1(t) = 0 \end{cases} . \tag{3.2}$$

Equation (3.2) is an holonomic constraint, it is integrable and can be embedded in to the system equations, and by that reduce the number of equations to six, that is the number of DoF .

In order to describe the orientation of the body's fixed frame F_r we will use the Roll-Pitch-Yaw (RPY) angles. RPY are three successive rotations of the body's fixed frame, all in respect to the inertial frame. Where Roll is the rotation about x_0, Pitch is the rotation about y_0 and Yaw is the rotation about z_0. Therefore F_b global angles are $\{\theta_1(t), \theta_2(t), \theta_3(t)\}$ about the corresponding $[x_b, y_b, z_b]^T$, and the translation of O_b is $[x(t), y(t), r]$, where r is the balls' radius. F_r orientation is described by the RPY rotations $\{\theta_4(t), \theta_5(t), \theta_6(t)\}$, and the translation of O_r is resulted of the robot's orientation and its relativeness to O_b.

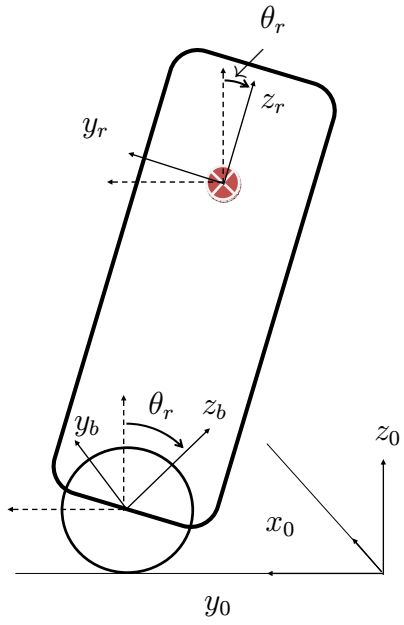


Figure 3.2: 2D representation of Coordinate systems used to describe the Roll or Pitch angles. The ball's frame y_b, z_b rotated in θ_b, and the body's fix frame y_r, z_r rotated in θ_r.

Denote $\tilde{q}$ to be the vector of the eight DoF ,

$$\tilde{q} = \begin{bmatrix} x(t) \\ y(t) \\ \theta_1(t) \\ \theta_2(t) \\ \theta_3(t) \\ \theta_4(t) \\ \theta_5(t) \\ \theta_6(t) \end{bmatrix}.$$

Under the definition of the holonomic non-slip constraint (3.2), the $x(t), y(t)$ DoF can be directly described as a function of the ball's rotations $\theta_1(t), \theta_2(t)$ about x_0, and y_0. The two constraints reduce the number of independent DoF to six.

Now we can define q as the vector of the six independent DoF . Further in this work, we keep a short writing by dropping the time depended notation $'(t)'$,

$$q = \begin{bmatrix} \theta_1(t) \\ \theta_2(t) \\ \theta_3(t) \\ \theta_4(t) \\ \theta_5(t) \\ \theta_6(t) \end{bmatrix} \Rightarrow \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix}.$$

Let us now build the kinematics of the whole Ballbot system. The homogeneous transformation matrix of the robot's fixed frame F_r, after the implementation of constraint 3.2 is:

$${}^0A_r = \begin{bmatrix} [{}^0R_r] & cm_r \\ 0_{1\times 3} & 1 \end{bmatrix} =$$

$$\begin{bmatrix} C\theta_5 C\theta_6 & S\theta_4 S\theta_5 C\theta_6 - S\theta_6 C\theta_4 & C\theta_4 S\theta_5 C\theta_6 + S\theta_6 S\theta_4 & cm_r\hat{x} \\ C\theta_5 S\theta_6 & S\theta_4 S\theta_5 S\theta_6 + C\theta_6 C\theta_4 & C\theta_4 S\theta_5 S\theta_6 - C\theta_6 S\theta_4 & cm_r\hat{y} \\ -S\theta_5 & S\theta_4 C\theta_5 & C\theta_4 C\theta_5 & cm_r\hat{z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where l is the distance between cm_r and cm_b, and we shortly notate $Sin(\theta_i)$ as $S\theta_i$ and $Cos(\theta_i)$ as $C\theta_i$. cm_r is given by:

$$cm_r = \begin{bmatrix} cm_r\hat{x} \\ cm_r\hat{y} \\ cm_r\hat{z} \end{bmatrix} = \begin{bmatrix} r\theta_1 + l\left(C\theta_4 C\theta_6 S\theta_5 + S\theta_4 S\theta_6\right) \\ -r\theta_2 + l\left(-C\theta_6 S\theta_4 + C\theta_4 S\theta_5 S\theta_6\right) \\ r + lC\theta_4 C\theta_5 \end{bmatrix}$$

In order to calculate the linear and angular velocities, two full Jacobian matrixes are calculated. One for the ball and one for the robot. The full Jacobian matrix is assembled of two parts, the linear Jacobian J_L, and the angular Jacobian J_ω. The linear Jacobian is assembled of the partial derivative of the $cm = [cm_{\hat{x}}, cm_{\hat{y}}, cm_{\hat{z}}]^T$ with respect to the DoF $q = [q_1 \dots q_6]^T$. The angular Jacobian is constructed of unit vectors, each in the direction of a rotational DoF , where their values are given with respect to the desired coordinate frame, in our case with respect to F_0. We notate the third element of a unit vector, pointing in the direction of q_2, represented with respect to F_0 as ${}^0q_{2,\hat{z}}$. Now the vector of linear velocity V, and the vector of angular velocity ω, are calculated as:

$$\begin{bmatrix} V \\ \omega \end{bmatrix} = [J]\dot{q}$$

And explicitly:

$$J = \begin{bmatrix} J_L \\ J_\omega \end{bmatrix} = \begin{bmatrix} \frac{\partial cm_x}{\partial q_1} & \frac{\partial cm_x}{\partial q_2} & \cdots & \frac{\partial cm_x}{\partial q_6} \\ \frac{\partial cm_y}{\partial q_1} & \frac{\partial cm_y}{\partial q_2} & \cdots & \frac{\partial cm_y}{\partial q_6} \\ \frac{\partial cm_z}{\partial q_1} & \frac{\partial cm_z}{\partial q_2} & \cdots & \frac{\partial cm_z}{\partial q_6} \\ {}^0q_{1,\hat{x}} & {}^0q_{2,\hat{x}} & \cdots & {}^0q_{6,\hat{x}} \\ {}^0q_{1,\hat{y}} & {}^0q_{2,\hat{y}} & \cdots & {}^0q_{6,\hat{y}} \\ {}^0q_{1,\hat{z}} & {}^0q_{2,\hat{z}} & \cdots & {}^0q_{6,\hat{z}} \end{bmatrix}$$

We remaind that the ball's linear velocity is described by (3.1), and its angular velocity is straightforward $\omega_b = (\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3)$. We now obtain the linear and angular velocities of the ball and the robot $V_b, \omega_b, V_r, \omega_r$. In this system the rotational Jacobians are

$$J_{\omega,r} = \begin{bmatrix} 0 & 0 & 0 & Cos(\theta_5)Cos(\theta_6) & -Sin(\theta_6) & 0 \\ 0 & 0 & 0 & Cos(\theta_5)Sin(\theta_6) & -Cos(\theta_6) & 0 \\ 0 & 0 & 0 & -Sin(\theta_5) & 0 & 1 \end{bmatrix} \tag{3.3}$$

$$J_{\omega,b} = [I_{3\times3}, 0_{3\times3}]\,.$$

3.2. Dynamics

In order to formulate the robots' equations of motion, partial derivatives of the Lagrangian according to the Euler-Lagrange equation are used. The closed form formulation of the matrix formed equation of motion is used [**26**]. The Lagrangian is the remainder of the overall

kinetic energy KE, which is velocities depended, and the potential energy PE, which is position and orientation depended. The ball and the robot kinetic energy are given as:

$$\begin{aligned} KE_b &= \frac{1}{2}[m_b V_b^T V_b + \omega_b^T I_b \omega_b] \\ KE_r &= \frac{1}{2}[m_r V_r^T V_r + \omega_r^T I_r \omega_r] \end{aligned} \tag{3.4}$$

Where m_b, m_r are the mass of the ball and the robot respectively. I_b, I_r are the tensor of inertia of the ball and the robot about their center of mass, represented with respect to F_0. The transformation of the tensor of inertia $\tilde{I}_r$ from its' principle axis to the inertial frame, is done using the rotation matrix:

$$I_r = {}^0R_r^T \, \tilde{I}_r \, {}^0R_r$$

We note that the ball is symmetrical about its center of mass, therefore its' tensor of inertia is a diagonal at all directions.

The potential energy of the ball and the robot are only gravitational

$$\begin{aligned} PE_b &= m_b \, g \, r \\ PE_r &= m_r \, g \, cm_z \end{aligned} \tag{3.5}$$

Based on equations (3.4) and (3.5) the Lagrangian is:

$$\begin{aligned} &L = KE - PE = \\ &\frac{1}{2}[m_b V_b^T V_b + \omega_b^T I_b \omega_b + m_r V_r^T V_r + \omega_r^T I_r \omega_r] - m_b \, g \, r - m_r \, g \, cm_z \end{aligned} \tag{3.6}$$

It is now possible to derive the equations of motion of the system, in their matrix form:

$$D\left(q\right)\ddot{q} + C\left(q, \dot{q}\right)\dot{q} + G\left(q\right) + Fr\left(\dot{q}\right) = \tilde{\tau} \tag{3.7}$$

Were $D(q)_{6\times6}$ is called the *inertia matrix*. It is symmetric and positive definite, and it is calculated as:

$$D(q) = m_b {J_{L,b}}^T J_{L,b} + {J_{\omega,b}}^T I_b J_{\omega,b} + m_r {J_{L,r}}^T J_{L,r} + {J_{\omega,r}}^T I_r J_{\omega,r}$$

$C(q,\dot{q})$ is the *Coriolis matrix*. So the term $C(q,\dot{q})\,\dot{q}$ comprises of the centripetal and coriolis terms. $G(q)$ is a vector of the gravitation effect on each DoF . $Fr(\dot{q})$ is a vector artificially added, in order to simulate the inner system's viscous friction. $\tilde{\tau}$ is the external generalized forces vector, presented in the direction of each DoF . In this system, all of the external generalized forces are outcomes of the torques applied between the robot's spherical actuator, which is fixed to the robot's body frame F_r, and the ball, which its DoF are relative to the inertial frame F_0. This torques need to be represented in three different ways. The first is as a three elements torque vector given with respect to F_r, this vector is the actual applied torques by the spherical actuator. The second, is the representation of this torques relative to F_b, in order to be compatible with the ball's DoF . The third is the representation in the direction of each of the robot's body DoF , in order to comply with the Euler-Lagrange formulation. The transformation between this representation is derived from the Euler-Lagrange formulation, as:

$$\tau = J^T \cdot \begin{bmatrix} F_{ext} \\ \tau_{ext} \end{bmatrix}, \tag{3.8}$$

Where F_{ext}, τ_{ext} are the external forces applied to the system, given with respect to the inertial frame. J^T is the Jacobian matrix of the system, given with respect to the same inertial frame. τ is the generalised forces vector presented in the direction of each of the system's DoF . For the

ballbot system (3.8) is in the form:

$$\tau = J_{\omega}^{T} \cdot \tau_{ext}. \tag{3.9}$$

As defined, τ_{ext} is the external torque applied to the system, represented in the inertial frame, in this case the ball's frame F_b, *i.e.* second representation. So we see that (3.9) define the relation between the representations of the same torques, in the direction of the ball's and the robot's body DoF .

In order to keep the simplicity of the controller formulation, we choose to work with the third representation as we define

$$\tau = \begin{bmatrix} \tau_R \\ \tau_P \\ \tau_Y \end{bmatrix}. \tag{3.10}$$

Where $\tau_{(3\times1)}$ is the three elements generalized torques vector - given in the direction of each DoF of the robot's body - RPY . In order to represent these elements in the direction of the ball's DoF , we follow (3.9) and yields with the total generalized forces vector

$$\tilde{\tau} = \begin{bmatrix} -J_{\omega,r}^{-T} . \tau_{(3\times1)} \\ \tau_{(3\times1)} \end{bmatrix}_{(6\times1)}, \tag{3.11}$$

Where $J_{\omega,r}$ is only the right 3×3 matrix of $J_{\omega,r}$ in Equation (3.3), and $J_{\omega,r}^{-T}$ is $\left(J_{\omega,r}^{T}\right)^{-1}$.

Next, we would like to calculate the actual torques applied by the spherical actuator, first representation. That is to represent the torque

vector elements with respect to the body's fixed frame F_r. In order to do so, we use the rotation matrix R_r and the torqe vector whose elements are given with respect to the ball's frame F_b.

$$\tau_{motor} = R_r^T \tau_b = R_r^T J_{\omega,r}^{-T} \tau. \tag{3.12}$$

The invertibility of $J_{\omega,r}$ depends only at θ_5, the Pitch angle of the robot's body, as can be seen in its Determinant

$$Det(J_{\omega,r}) = Cos(\theta_5). \tag{3.13}$$

It is obvious that $J_{\omega,r}$ is singular for $\theta_5 = \pm 90^\circ$. It makes sense since in that position the robot is horizontal, and oriented in a way that its actuator's Roll and Yaw directions are linearly dependent under the RPY definition. This singularity will further effect the controller development, with the same bound.

We remaind here again that this system is underactuated, as it is clear now that the system has only three actuator torques for the six independent DoF .

We note that we constrict the model to keep a continues smooth functions, as we assume the non-slip constraint between the spherical actuator and the ball. Modeling the static and kinetic friction occurring in the system is a wide problem, and will be modestly referred in this work.

CHAPTER 4

Controller Design

In this chapter a non-linear control law is derived. The controller's goal are two folds. First, to stabilize the robot in its upright naturally unstable equilibrium point. Second, to position the robot at the desired location on the horizontal ,$x - y$, plane and the body rightly pivoted about z_0. Due to the underactuated nature of the system, and more particularly - a low actuated system [**35**], a partial feedback linearization control is derived based on [**29, 31, 35, 36**], after manipulating the equations of motion to a specific form. By doing so we achieve global exponential stability of the robot's body only $\{\theta_4, \theta_5, \theta_6\}$. The stabilization controller may cause the system a drift of the first three DoF , *i.e.* the ball's, which under the no-slip constraint are proportional to the robot position. In order to compensate the drift, a position control law is added. The overall system's stability, considering both stabilization and position controllers, is proven for local stability by using *Lyapunov's indirect method* [**19**, p.139]. The control law is designed assuming a full state space feedback.

4.1. Partial Feedback Linearization

The first step in the controller design is to derive the stability controller, whose purpose is to bring the robot's body up vertically and keep keep it stabilized.

LEMMA 4.1. *Let the equations of motion of the underactuated system to be of the form:*

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} -{J_\omega}^{-T}\tau \\ \tau \end{bmatrix}, \tag{4.1}$$

Where $q_1, q_2 \in I\!R^{n\times 1}$ *are the vectors of independent DoF of the system,* $D_{i,j} \in I\!R^{n\times n}$; $i,j \in (1,2)$, $s_1, s_2 \in I\!R^{n\times 1}$, $\tau \in I\!R^{n\times 1}$ *is the external control vector expressed correspondingly to* q_2 *DoF and* ${J_\omega}^{-T}$ *is the inverse of the transposed jacobian matrix presents the transformation of coordinates between* q_1 *and* q_2.
Denote:

$$\begin{aligned} D_4 &= D_{22} - D_{21}\left({J_\omega}^T D_{11} + D_{21}\right)^{-1}\left({J_\omega}^T D_{12} + D_{22}\right) \\ s_4 &= s_2 - D_{21}\left({J_\omega}^T D_{11} + D_{21}\right)^{-1}\left({J_\omega}^T s_1 + s_2\right). \end{aligned} \tag{4.2}$$

Assuming there exist $\left({J_\omega}^T D_{11} + D_{21}\right)^{-1}$ *and* ${D_4}^{-1}$, *and* q_2^d *is the desired* q_2 *for time* $t \to \infty$, *then the equilibrium point* $q_2 = q_2^d$ *is globally exponentially stable, under the control input:*

$$\tau = D_4 u_{stab} + s_4, \tag{4.3}$$

Where $u_{stab} = -K_p(q_2 - q_2^d) - K_d\dot{q}_2$ *is the partial linearized feedback,* $K_p = diag(k_{p1} \dots k_{pn}) \in I\!R^{n\times n}$ *and* $K_d = diag(k_{d1} \dots k_{dn}) \in I\!R^{n\times n}$, *for all* $0 < k_{pi}$ *and* $1 < k_{di}$.

PROOF. Sub-divide equation (4.1) into two equations yield:

$$J_\omega{}^T \left(D_{11}\ddot{q}_1 + D_{12}\ddot{q}_2 + s_1\right) = -\tau \tag{4.4}$$

$$D_{21}\ddot{q}_1 + D_{22}\ddot{q}_2 + s_2 = \tau. \tag{4.5}$$

The summation of (4.4) and (4.5) produce the following equivalent dynamical system:

$$\begin{aligned} D_{31}\ddot{q}_1 + D_{32}\ddot{q}_2 + s_3 &= 0 \\ D_{21}\ddot{q}_1 + D_{22}\ddot{q}_2 + s_2 &= \tau \end{aligned} \tag{4.6}$$

where for convenient we define:

$$\begin{aligned} D_{31} &= \left(J_\omega{}^T D_{11} + D_{21}\right) \\ D_{32} &= \left(J_\omega{}^T D_{12} + D_{22}\right) \\ s_3 &= J_\omega{}^T s_1 + s_2. \end{aligned}$$

Isolating $\ddot{q}_1$ from (4.6) gives:

$$\ddot{q}_1 = -D_{31}{}^{-1}(D_{32}\ddot{q}_2 + s_3). \tag{4.7}$$

Notice that under Lemma 4.1 assumption D_{31} is invertible.

We now substitute (4.7) into (4.5) and collect elements as follows:

$$D_4\ddot{q}_2 + s_4 = \tau, \tag{4.8}$$

Where

$$D_4 = D_{22} - D_{21}{D_{31}}^{-1}D_{32}$$

$$s_4 = s_2 - D_{21}{D_{31}}^{-1}s_3.$$

As define in (4.3) and under Lemma's 4.1 assumptions D_4 is invertible, so the control law τ is set to be:

$$\tau = D_4 u_{stab} + s_4$$

where

$$u_{stab} = -K_p\left(q_2 - q_2^d\right) - K_d\dot{q}_2.$$

The substitution of (4.3) into (4.7) and (4.8), enables us to represent the closed loop system in the form:

$$\ddot{q}_1 = -{D_{31}}^{-1}\left(D_{32}\left(-K_p\left(q_2 - q_2^d\right) - K_d\dot{q}_2\right) + s_3\right) \tag{4.9}$$

$$\ddot{q}_2 = -K_p\left(q_2 - q_2^d\right) - K_d\dot{q}_2 \tag{4.10}$$

Equation (4.10) is actually a set of independent linear ordinary non-homogenous differential equations, with positive gains. Each q_{2_i} equation is exponentially converge to zero, as proven in [**25**, p.192] □

Although q_2 is proven to be globally exponentially stable, this controller may cause a drift in q_1. In order to compensate this drift, a simple P.D controller is added, and will be referred as the position controller.

4.2. Position Control

As mentioned in the chapter's prologue, the stabilization controller derived in section 4.1, may cause a drift in q_1, a simple P.D. position controller is added, using the form:

$$u_{pos} = K_{pos}\left(q_1^d - q_1\right) - K_v\dot{q}_1, \tag{4.11}$$

where K_{pos}, K_v are diagonal matrices with positive gains. q_1^d is the desired position vector as the first two elements of q_1 are proportional to the position of the ball's center, under the non-slip constraint (3.2). The overall control law is achieved by the adding (4.11) to (4.3) and result in:

$$\tau = D_4 u_{stab} + s_4 + u_{pos} \tag{4.12}$$

This addition to the controller slightly disturb the global exponential stability achieved by Lemma 4.1. The new system is proven to be locally stable. The proof is based on two steps. In the first step we make use of Lyapunov indirect method and check the locus of the poles of the closed loop linearized system. Then we notice that almost all the poles are located in the OLHP *i.e.* stable poles, there is one pole at the origin. That pole holds the system's stability undetermined. The undetermined pole is associated with the ball's rotation about z_0, as can be seen in its associated eigenvector. We use the following Lemma 4.2, in order to derive the conditions for witch the system can be disassembled into two sub-systems. One holds the stable poles and therefore asymptotically stable, and the other holds the undetermined pole, zero, and therefore undetermined. Under the Lemma conditions the full system is proven to be stable.

LEMMA 4.2. *Let two dynamical systems be of the form:*

$$\dot{w}_1 = f_1(w_1(t)) \tag{4.13}$$

$$\dot{w}_2 = M w_1(t) \tag{4.14}$$

where $w_1(t) \in \mathbb{R}^n, w_2(t) \in \mathbb{R}^m$, $f_1(w_1(t))$ *is smooth, twice continuously differentiable and bounded in a neighborhood* D *containing the origin. Assume* $\|f_1(0)\| = 0 \quad \forall t_0 \leq t$ *is an equilibrium point of* $f_1(w_1)$, *and* $M \in \mathbb{R}^{m \times n}$ *is a constant matrix. Let the linear approximation of* $f_1(w_1)$ *be*

$$\dot{w}_1 \approx A w_1 \quad \text{where} \quad A = \left.\frac{\partial f_1(w_1)}{\partial w_1}\right|_{w_1=0}, \tag{4.15}$$

Denote $\lambda_i(A)$ *for* $i \in \{1 \dots n\}$, *to be the eigenvalues of* A.
If all $Re(\lambda_i(A)) < -b$ *for* $0 < b \in \mathbb{R}$, *then there exist a neighborhood* $U \subset D$ *containing the origin such that* $w_1(t=0) \in U$, (4.14) *is stable, and*

$$\lim_{t \to \infty} (\|w_2(t)\|) \leq w_2^* \tag{4.16}$$

where $w_2^* \in \mathbb{R}^m$ *is a constants vector.*

PROOF. We assume all $Re(\lambda_i(A)) < -b$ for $0 < b \in \mathbb{R}$, then by following [**15**, p.180] as detailed in Appendix A, there exist a neighborhood $U \subset D$ containing the origin such that for $w_1(t=0) \in U$ system (4.13) is locally exponentially stable, hence,

$$\|w_1(t)\| \leq e^{-tc} \|w_1(t=0)\| \quad \forall \ \ w_1(t=0) \in U, \tag{4.17}$$

where $0 < c < b \in I\!R$. Because $w_1(t)$ converge exponentially to its equilibrium point , $w_2(t)$ exponentially converge as well. That is

$$\|\dot{w}_2\| = \|Mw_1(t)\| \leq e^{-tc} \|Mw_1(t=0)\| = \beta e^{-ct},$$

for $0 < \beta \in I\!R$. We note that in order to maintain simplicity and without lose of generality, we can choose $t_0 = 0$. So that

$$\|w_2(t)\| = \left\| \int_0^t \dot{w}_2(t)dt \right\| \leq \int_0^t \|\dot{w}_2(t)\| \, dt \leq \int_0^t \beta e^{-ct} dt = -\frac{\beta e^{-ct}}{c} + \frac{\beta}{c}$$

hence $w_2(t)$ is stable as

$$\lim_{t\to\infty} (\|w_2(t)\|) \leq \frac{\beta}{c} = const \tag{4.18}$$

□

We now hold a proof for the existence of a neighborhood of the equilibrium point, that ensure the system's stability if its initial conditions contained in that neighborhood. Next we implement Sections 4.1 and 4.2 to the Ballbot system that was derived in Chapter 3.

4.3. Implementation of Control Design

In this section we present the implementation and use of the overall controller design to the Ballbot system, as detailed in sections 4.1 and 4.2.

First we establish Section 4.1. We note that for the Ballbot system, $D_{i,j} \in I\!R^{3\times 3}$ for $i, j \in (1, 2)$, are block matrices of the inertia matrix D

which is symmetric and positive definite. $s_1, s_2 \in I\!R^{3\times 1}$ are the corresponding parts of the vectors $C\left(q,\dot{q}\right)\dot{q}$ and $G(q)$, as $C\left(q,\dot{q}\right)\dot{q}$ holds the Coriolis and Centripetal forces terms and $G(q)$ holds the torques applied by gravitation. $\tau \in I\!R^{3\times 1}$ is the external control vector, expressed in the direction of each of the robot's body DoF .

As define in Lemma's 4.1 assumptions, the current D_{31} is indeed invertible for all $[-90^\circ < \theta_5 < 90^\circ]$. It is clearly shown in the simplified expression of its determinant,

$$Det(D_{31}) = ib \cdot C\theta_5 \left(ib + (mb + mr)r^2 + l \cdot mr \cdot C\theta_4 C\theta_5\right)^2 . \quad (4.19)$$

Where mb, mr are the the ball's and the robot's mass, r is the ball's radius, l is the distance between the ball's center to the body center of mass and ib is the moment of inertia of the ball, and is given as

$$ib = \frac{2}{5} mr \cdot r^2.$$

Due to the complexity of the analytical derivation for the determinant of D_4, it is hard to establish direct dependency of its inversion feasibility in the system's variables. Instead, we establish the invertibility assumption of D_4 by preforming a numerical evaluation of its determinant for the extreme ranges of the angles $[0^\circ \leq \theta_4 \leq 70^\circ]$, $[0^\circ \leq \theta_5 \leq 70^\circ], [0^\circ \leq \theta_6 \leq 360^\circ]$, and the estimated system's dimensions. This range of angles, produce a global angle of the robot's body off the inertial vertical axis Z_0 up to 83°. The calculation of the global angle considering Figure 4.1, is as follows:

$$CM_r = r + l \cdot Cos(Glob\theta_z),$$

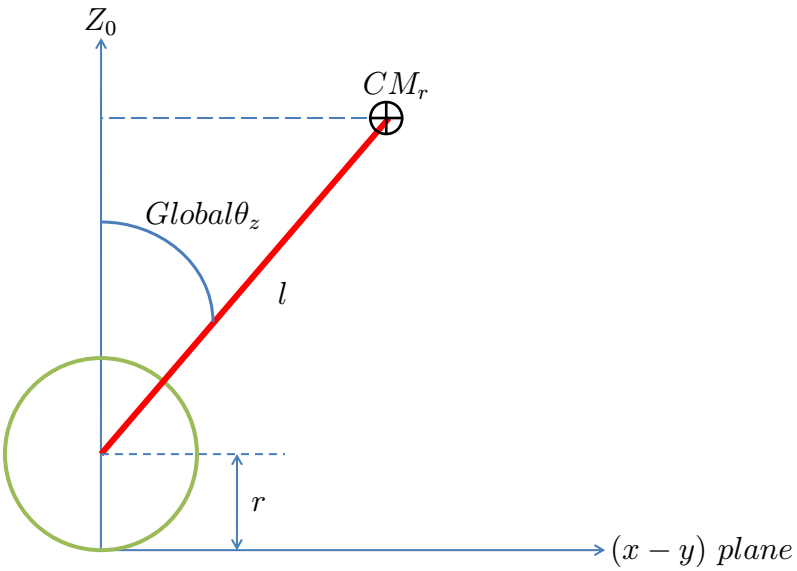


Figure 4.1: $Global\theta_z$ calculation param5eters

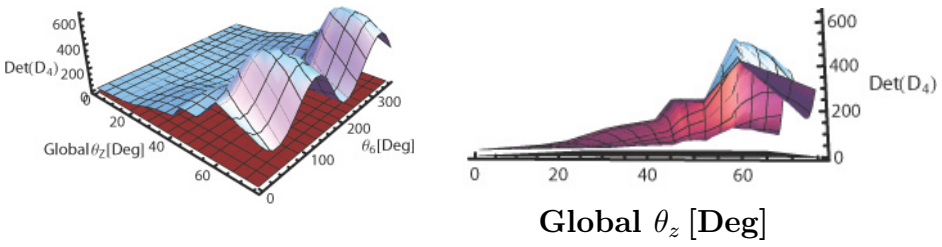


Figure 4.2: Numerical evaluation of D_4 determinant in the ranges: $[0° \leq Global\,\theta_z \leq 80°]$, $[0° \leq Yaw \leq 360°]$

and therefore

$$Glob\theta_z = ArcCos\left(\frac{CM_r - r}{l}\right).$$

Figure 4.2 presents the numerical evaluation of the determinant of D_4 with respect to $Global\theta_z$, and the robot's Yaw angle θ_6.

As we validate all Lemma's 4.1 assumptions, we can assure the stabilization controller for q_2.

Next, following Section 4.2, we wish to prove the closed loop system, including the position control law as stable. In order to do so, we need to prove the tenability of the system to Lemma's 4.2 requirements.

The first step is to reduce the system's order by bringing the closed loop dynamic equations into a first order form, *i.e.* state space

$$\dot{x} = f(x), \tag{4.20}$$

with the corresponding state vector $x = (\theta_1 \dots \theta_6, \dot{\theta_1} \dots \dot{\theta_6})^T \in I\!R^{12}$.

Next, we approximate the state space by preforming a linearization about the equilibrium point $\bar{x}$

$$\begin{aligned} \dot{x} &\approx Ax, \\ A &= \left.\frac{\partial f(x)}{\partial x}\right|_{x=\bar{x}}. \end{aligned} \tag{4.21}$$

Now we investigate the dynamic behavior of the linear system by, examine the eigenvalues of the matrix A in (4.21). The dependency of

the eigenvalues in the controller gains is given by

$$\begin{aligned}
\lambda_1 =& 0 \\
\lambda_{2,3,4} =& Root(c_{11}kpy + E(c_{12} + c_{11}kdy + kpy) + E^2(c_{13} + kdy) + E^3) \\
\lambda_{5,6,7,8} =& Root(c_{21}kposp + E(c_{22} + c_{23}kposp + c_{21}kvp) + \\
& E^2(c_{24} + c_{25}kposp + kpp + c_{21}kvp) + E^3(c_{26} + kdp + c_{25}kvp) + E^4) \\
\lambda_{9,10,11,12} =& Root(c_{21}kposr + E(c_{22} + c_{23}kposr + c_{21}kvr) + \\
& E^2(c_{24} + c_{25}kposr + kpr + c_{21}kvr) + E^3(c_{26} + kdr + c_{25}kvr) + E^4)
\end{aligned} \tag{4.22}$$

Where E is a "dummy" variable used to notate the polynomial expression, the gains of the controller's diagonal matrices are:

$$\begin{aligned}
K_p &= Diag\,(kpr, kpp, kpy) \\
K_d &= Diag\,(kdr, kdp, kdy) \\
K_{pos} &= Diag\,(kposr, kposp, kposy) \\
K_v &= Diag\,(kvr, kvp, kvy)\,.
\end{aligned}$$

and all of the coefficients $c_{i,j}$ are terms of the system's dimensions *i.e.* lengths, masses and moments of inertia.

As known from control theory [**19**, p.139] and [**15**, p.180], in order to achieve local exponential stability of system (4.20), we must demand

$$Re\,(\lambda_i(A)) < -b \quad ; \quad 0 < b \in I\!R. \tag{4.23}$$

Prima facie, complying with equations (4.23) is impossible because of $\lambda_1 = 0$. Reinvestigation of (4.20) shows that it is independent of the state variable x_3. Further more, the eigenvector correspond to the

eigenvalue $\lambda_1 = 0$, is in the direction of x_3, $v_1 = (0, 0, 1, 0 \dots)^T$. It is stem that because x_3 is a passive state variable, we can separate its equation from the overall dynamics. We now obtain two dynamical systems of the form (4.13)(4.14), where

$$w_1 = \tilde{x} \quad ; \quad \tilde{x} = x_i, i = \{1, \dots, 12\}, i \neq 3$$

$$w_2 = x_3,$$

and we now study them separately as in Lemma 4.2.

First we need to comply with Lemma's 4.2 definitions. By the right selection of the controller gains, we can satisfy (4.23) for $f_1(w_1)$. According to [**15**, p.180] and as detailed in Appendix A, there exist a neighborhood U containing the origin such that $w_1(t = 0) \in U$ and $f_1(w_1)$ is exponentially stable. This satisfies Lemma's 4.2 definitions for $f_1(w_1)$ exponential converges. Now, in our system $w_2 = x_3$ so that $\dot{w}_2 = \dot{x}_3 = Mx$, as $M \in \mathbb{R}^{1\times 11}$ is all zeros accept 1 in the ninth place.

That proves the overall system w_1 and w_2 as stable.

Equation (4.22) is actually a controller design tool. It gives us the parameters in order to validate (4.23) by the right selection of controller gains.

Considering the system's dimensions as specified for the simulation - Chapter 5, (4.22) correspond to:

$$\begin{aligned}
\lambda_1 =& 0, \\
\lambda_{2,3,4} =& Root[0.491464kpy + (0.127653 + 0.491464kdy + kpy)E \\
&+ (1.56097 + kdy)E^2 + E^3], \\
\lambda_{5,6,7,8} =& Root[401.824 + 101.312E + (-56.844 + kpp)E^2 \\
&+ (-14.3398 + kdp)E^3 + E^4], \\
\lambda_{9,10,11,12} =& Root[401.824 + 101.312E + (-56.844 + kpr)E^2 \\
&+ (-14.3398 + kdr)E^3 + E^4]
\end{aligned} ,$$

and for the selected controller gains, yields the eigenvalues

$$\begin{aligned}
&\lambda_1 = 0 && , \lambda_2 = -0.494756, \\
&\lambda_3 = -4.09789 && , \lambda_4 = -0.495423, \\
&\lambda_5 = -9.63493 - 12.9825i && , \lambda_6 = -9.63493 + 12.9825i, \\
&\lambda_7 = -0.206037 - 1.34853i && , \lambda_8 = -0.206037 + 1.34853i, \\
&\lambda_9 = -9.63493 - 12.9825i && , \lambda_{10} = -9.63493 + 12.9825i, \\
&\lambda_{11} = -0.206037 - 1.34853i && , \lambda_{12} = -0.206037 + 1.34853i
\end{aligned} .$$

It is now clear that in order to hold requirement (4.23) for the other eleven coupled equations, a right selection of the controller gains is needed. The dependency of the eigenvalues on the controller gains as presented in (4.22), provide a tuning tool for the controller.

A close look at (4.22) reveals two interesting properties. First, after the linearization the controller's gain matrixes elements has an independent impact on the system's eigenvalues, correlated to their RPY directions. Second, due to the system symmetry, the constants c_i are the same for Roll and Pitch direction. This properties are good for the control designer, as it is easier to compute the gains independently for each direction.

In summary of this section. At the beginning we validate that the stabilization control can be applied to the system. Then, in order to overcome the drift in q_1, we added a q_1 dependent P.D controller. In order to prove the new closed loop system stability, we used the dependency of the stability of the nonlinear system on the eigenvalues of matrix A of the linearized system. Next we expressed the dependency of the eigenvalues on the controller gains, and used Lemma 4.2 in order to separate the unstable eigenvalue. Finally, according to Lemma 4.2, the overall system is proven to be stable since (4.16) is validated for the system.

We note that (4.14) can be understood as a non-holonomic constraint between w_2 and w_1, and in some case can imply about the characteristic behavior of the system.

In spite of the humble basin of attraction presented in this section, as it derived out of the linear approximation of the system. The non linear controller capability, handling wide range of initial conditions is later presented in Chapter 5.

4.4. Friction and Dissipation Consideration

After the derivation of the control law, we mention the friction function $Fr(\dot{q})$, artificially embedded into the system's dynamic equations (3.7). The kinematic and the followed dynamic models of the system, premise a merely geometrical conditions. In particular, the tangent geometrical point of contact between the ball and the horizontal plane, and between the ball and the robot. Assuming the no-slip constraint at this points ease the modelling process. This constraint, as described in (3.2), is an holonomic constraint, so it is integrable and can be substitute into the kinematic and dynamic equation, and by that reduce the number of equations. The dynamic interpretation of the no-slip constraint is equivalent to an unbounded static friction force acting at the point of contact. At the ball-robot contact point, it allows us to assume the form of the torque vector (3.11). At the ball-plane contact point, it enable us to describe the relation between the ball's position on the horizontal plane to its DoF (3.2). Pivoting of the ball, as it is the rotation about the vertical z_0 axis, does not fall under the no-slip assumption because it cause no horizontal progression of the robot. Moreover, as the dynamic model assume a geometrical tangent point of contact, no friction torque can be referred to that point as it has zero lever arm about z_0. We now face a problem, the ball's Yaw DoF is forced by the spherical actuator at one side, and has no counter torque to balance the angular momentum on the other side. After orientating the robot's Yaw angle as desired, under the lack of control to the ball's Yaw angle and due to the conservation of angular momentum

in the absents of external torque, the ball will endlessly rotate the in Yaw direction, *i.e.* the system is not stable.

An engineering understanding of the real world robot system comes as a great help. Physical bodies are not absolutely rigid as they are not utterly round or flat. So the point of contact between the ball and the horizontal plane can be considered as a small circular area. This realization of the system, allow us to ascribe a viscous friction torque about the ball's Yaw axis, and by that provides the external counter torque to the angular momentum equation.

The added friction is modeled as a simple viscous friction, *i.e.* proportional to the angular velocity. The gain is scaled out of [**39**]. The system response under the absent of the added friction is shown in subsection 5.1.2.

4.5. Linearized Ballbot System

This section presents the linearized dynamical system, following the derivation preformed in Chapter 3. We aim to assessment the assumption that the DoF are decoupled in the linear model. The linearization is preformed as in Equation (4.15). The parametric expressions are too long to be presented here, so the numerical values of the system are presented, after implementing the system dimensions and controller gains as detailed in Tables 5.1 and 5.2. The actual values are not as

important as the matrix structure.

$$A = \left[\begin{array}{cccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \mathbf{a} & 0 & 0 & \mathbf{b} & 0 & 0 & \mathbf{c} & 0 & 0 & \mathbf{d} & 0 & 0 \\ 0 & \mathbf{a} & 0 & 0 & \mathbf{b} & 0 & 0 & \mathbf{c} & 0 & 0 & \mathbf{d} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{i} & 0 & 0 & \mathbf{j} & 0 & 0 & \mathbf{k} \\ \mathbf{e} & 0 & 0 & \mathbf{f} & 0 & 0 & \mathbf{g} & 0 & 0 & \mathbf{h} & 0 & 0 \\ 0 & \mathbf{e} & 0 & 0 & \mathbf{f} & 0 & 0 & \mathbf{g} & 0 & 0 & \mathbf{h} & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{l} & 0 & 0 & \mathbf{m} & 0 & 0 & \mathbf{p} \end{array}\right] \tag{4.24}$$

where $a = 68.7497, b = 3748.2, d = 20.326, e = 432.023, f = -6.36607,$

$g = -340, h = -1.88214, i = -40.0079, j = 794.698, k = -1.47439,$

$l = 227.056, m = -70, n = 0.0865801, p = -20.0866.$

As we can see in Equation (4.24) the linearized matrix holds the following form: The upper six rows are formed due to the order reduction of the system into a first order state space, and the lower six rows are the linearized system's dynamics. Due to the symmetry of the system we see the diagonal repeating of values in the Roll and Pitch directions. More significantly, the values in each row are non zero every third column, meaning that it multiples only the values in

the same directions. The sevens row for example, **{a,b,c,d}** multiplies $\{\theta_1, \theta_4, \dot{\theta_1}, \dot{\theta_4}\}$ respectively, all are at the same direction - when discarding the other rotations.

That means that in a close neighborhood of the origin, linear approximation of the spatial system as two independent unicycles, can be considered. Further, regarding the CMU model of Ballbot, we see that we have additional two equations that are relevant to the ball's and body's rotations about z_0.

CHAPTER 5

Simulations and Experiments

Based on the dynamic model develop in chapter 3 and the constructed control law in chapter 4, a computer simulation is obtained and the results are presented. Experiments were made using a limited scaled down system. The experimental system is build from the LEGO NXT MindStorm kit, while the programming was done in designated environment for Matlab Simulimk, in order to communicate with the LEGO kit.

5.1. Simulations

5.1.1. Simulation Definitions. In order to check and diagnose the designed controller before building the full mechanical system, a computer code simulating the controller environment has to be established. The kinematic and dynamic equations of the system are formulated as described in chapter 3. The next step is apply the designed controller to the dynamic model, by substituting the feedback control function τ_{tot} into the dynamic equations. As expected, we now obtain a set of second order non-liner differential equations, whose variables are the system's DoF . The solutions of these equations describe the behavior of the system's DoF in time. Because an analytical solution

cannot be obtained, a numerical solution is preformed, based on the initial conditions of the DoF .

Performing the computational simulations, provides the convenience of examine the change in the system's behavior, as a result of changes in its variables and initial conditions. In order to perform the numerical simulation the ball's and the robot's dimensions, mass and moment of inertia has to be determined. Because non full sized system where built, this parameters ware scaled out of [**38**] and by approximating the Ballbot [**3**] using a CAD software. Details are presented in Table 5.1. We note that the gravity acceleration g is along the inertial axis z_0, $\tilde{I}_r$ values are given with respect to the robot's principle axes, and due to the ball's symmetry its tensor of inertia is invariant to its frame orientation. The simulation code was written using Walfram-Mathmatica© versions 7. The differential equation are solved using the built in *NDSolve* function for all 6 DoF and their first time derivative, for the time span of 10 seconds. The built in function calculation consider the machine precision of 15 digits, as well as adaptive integration step size. The step size is determined by keeping the solution error estimation within the tolerance of the machine precision.

Table 5.1: System's Parameters

Ball's radius	r	=	$0.11\ [m]$
Ball's mass	m_b	=	$3.12\ [Kg]$
Robot's mass	m_r	=	$42.04\ [Kg]$
Gravity acceleration	g	=	$9.81\ \left[\frac{m}{sec^2}\right]$
Distance of the robot's cm from the ball's cm	l	=	$0.95\ [m]$
Ball's Tensor of Inertia	I_b	=	$\begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{bmatrix} [Kg \cdot m^2]$
Robot's Tensor of Inertia	$\tilde{I}_r$	=	$\begin{bmatrix} 12.05 & 0 & 0 \\ 0 & 12.05 & 0 \\ 0 & 0 & 2.31 \end{bmatrix} [Kg \cdot m^2]$
Viscous Friction Coefficients	$fric$	=	$\begin{bmatrix} 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0 & 0 & 0 \\ -0.2 & 0 & 0 & 0.2 & 0 & 0 \\ 0 & -0.2 & 0 & 0 & 0.2 & 0 \\ 0 & 0 & -0.2 & 0 & 0 & 0.2 \end{bmatrix} \left[\frac{Kg \cdot m^2}{sec}\right]$

Gains of the control law were tuned using try and error procedure, presented in Table 5.2.

Table 5.2: Control gains values

Feedback Linearization - Proportional gains	K_p	=	$\begin{bmatrix} 340 & 0 & 0 \\ 0 & 340 & 0 \\ 0 & 0 & 70 \end{bmatrix}$
Feedback Linearization - Differential gains	K_d	=	$\begin{bmatrix} 40 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 20 \end{bmatrix}$
Position Control - Proportional gains	K_{pos}	=	$\begin{bmatrix} 23 & 0 & 0 \\ 0 & 23 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Position Control - Differential gains	K_{vel}	=	$\begin{bmatrix} 7 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

5.1.2. Non added Friction Simulation. In this subsection we present the system's response while discarding the added friction. The main influence to the system is to the ball's Yaw angle, as it is the only uncontrolled DoF . The initial conditions are as in (5.4):

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix}_{(t=0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 5^\circ \\ 5^\circ \\ 0 \end{bmatrix}$$

$$\dot{\theta}_{1(t=0)} = \dot{\theta}_{2(t=0)} = \dot{\theta}_{3(t=0)} = \dot{\theta}_{4(t=0)} = \dot{\theta}_{5(t=0)} = \dot{\theta}_{6(t=0)} = 0$$

As detailed in section 4.4, the drift of the ball's Yaw angle is clear in Figure 5.1e.

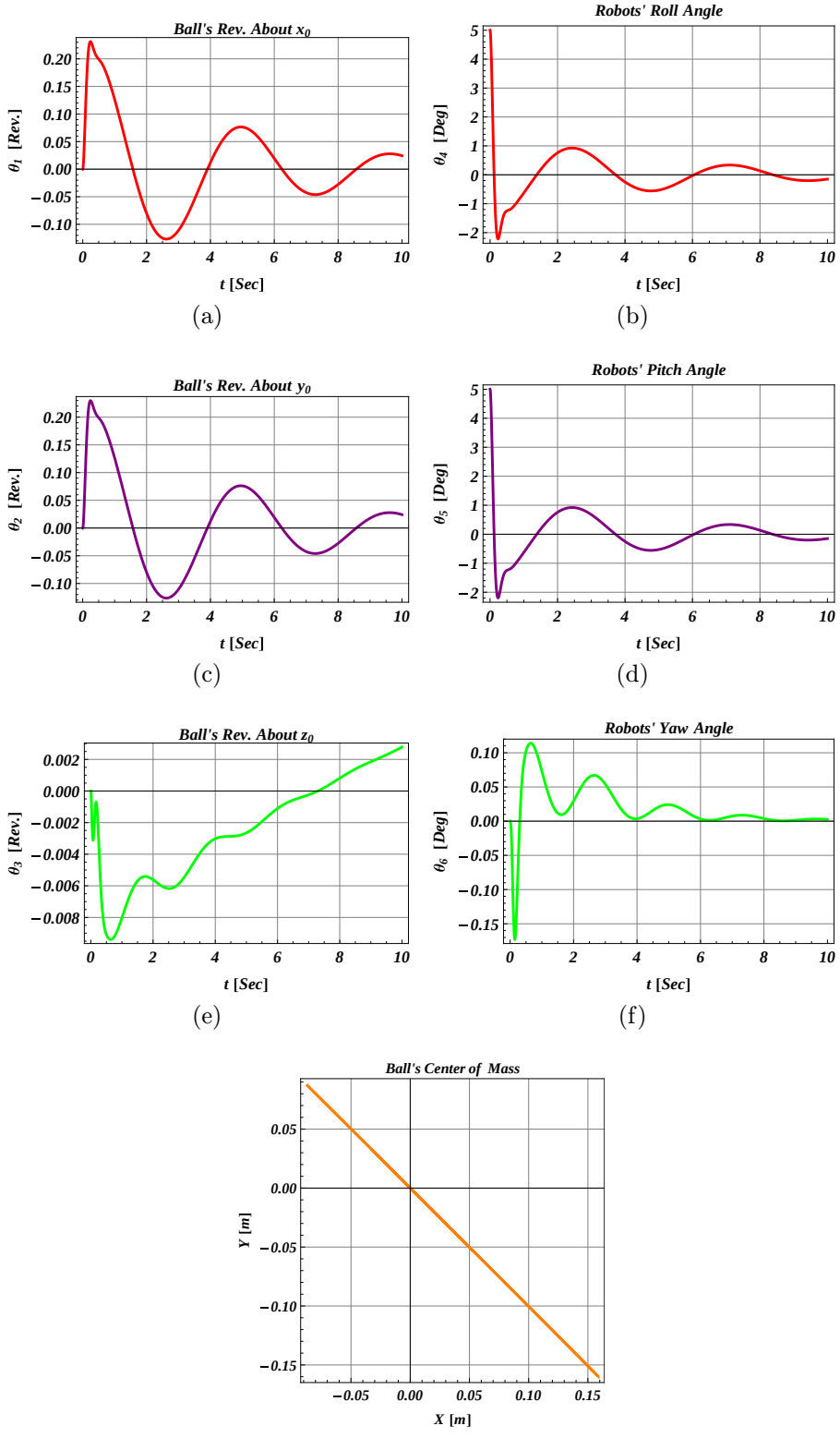


Figure 5.1: Simulation results of the ball's revolutions, the robot's RPY angles and plane travel, considering no added friction, for the I.C. (5.4)

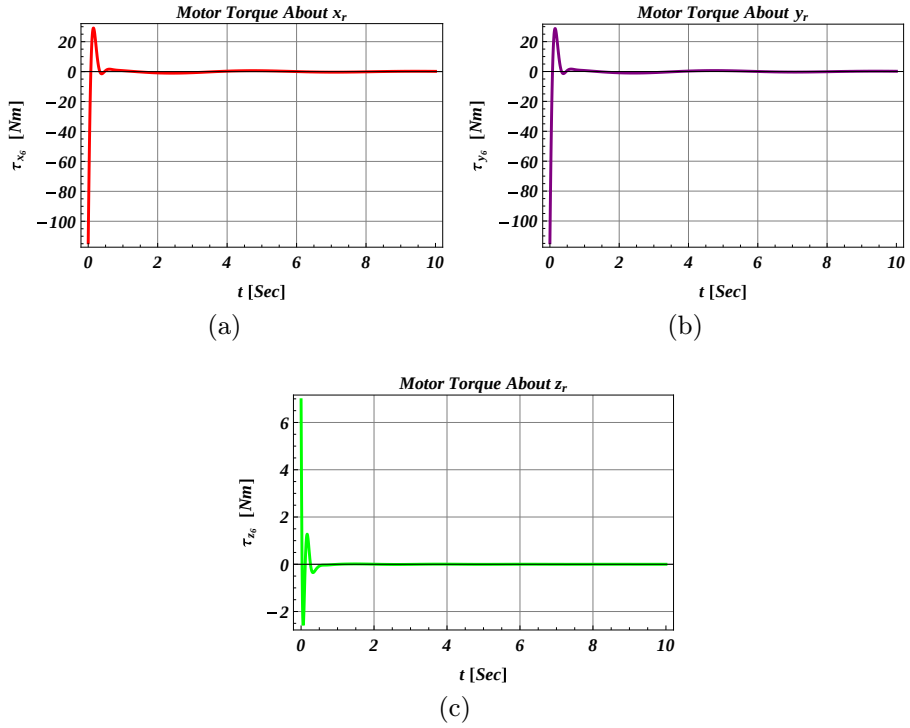


Figure 5.2: Simulation results of the RPY torques, considering no added friction, for the I.C. (5.2).

5.1.3. Stabilization Without Position Control. At first we present the performance of the stabilization control only, as derived in section 4.1. The simulation is done considering initial conditions of zero velocities, and angels as follows:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix}_{(t=0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 15^\circ \\ 15^\circ \\ 0 \end{bmatrix} \tag{5.1}$$

$$\dot{\theta}_{1(t=0)} = \dot{\theta}_{2(t=0)} = \dot{\theta}_{3(t=0)} = \dot{\theta}_{4(t=0)} = \dot{\theta}_{5(t=0)} = \dot{\theta}_{6(t=0)} = 0$$

The combination of these angles orient the robot's body in a global angle of about 21° relative to the inertial vertical axis z_0. These angle is considered as the upper boundary limit, where linear approximation can be assumed good.

Figure 5.3 presents the expected exponential converges of q_2 to equilibrium point despite the relatively large initial conditions of 5.1. As expected the ball's yaw angle does not converge to 0, as it is not controlled. Figures 5.3a and 5.3c display the drift for the Roll and Pitch angles of the ball, as the drift of Yaw angle ebb due to the added friction. Figure 5.3g presents the travel of the ball on the horizontal ,$x-y$, plane.

Figure 5.4 presents the corresponding torques applied in order to achieve stability.

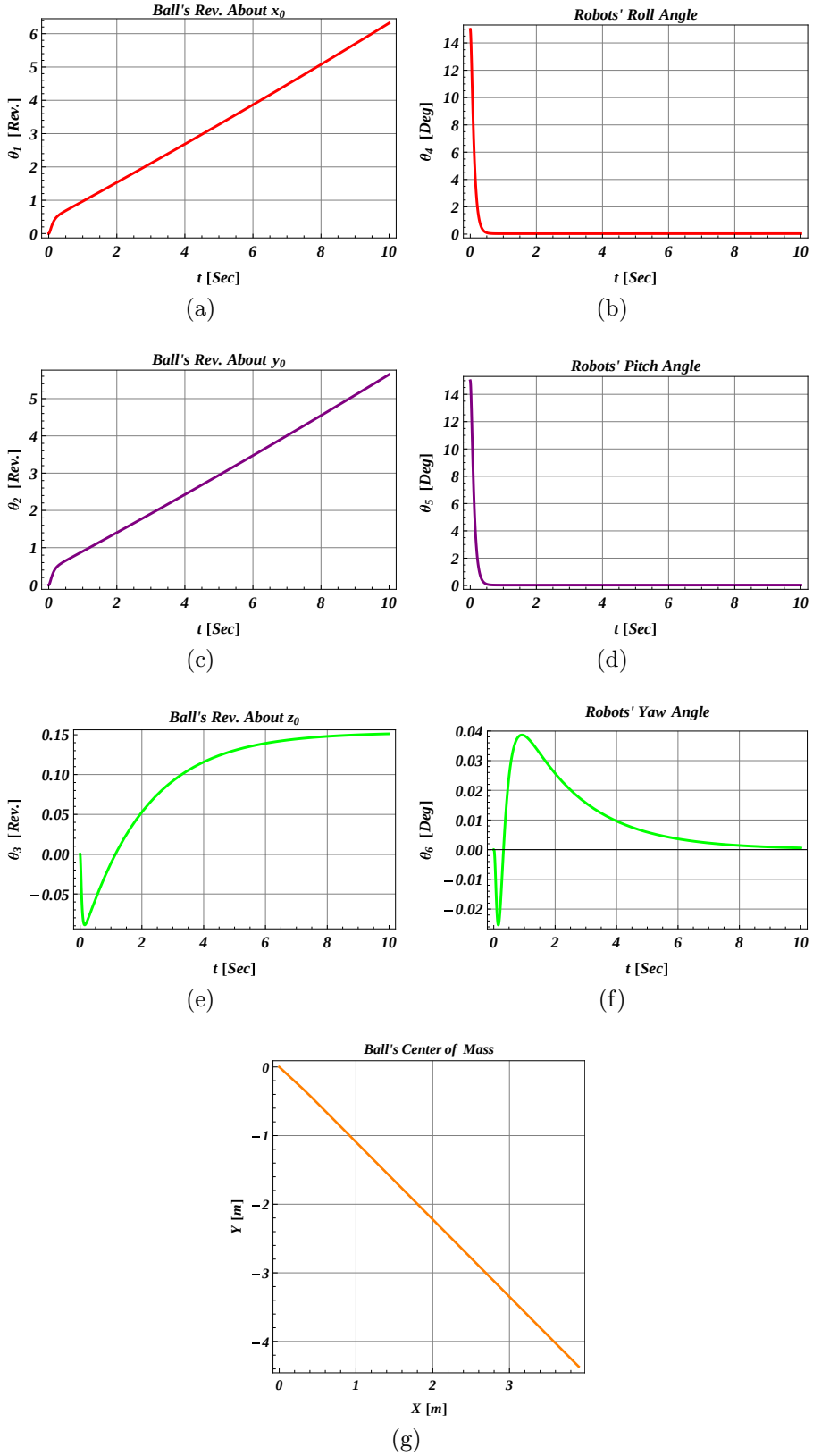


Figure 5.3: Simulation results of the ball's revolutions, the robot's RPY angles and plane travel, considering no added friction, for the I.C. (5.1)

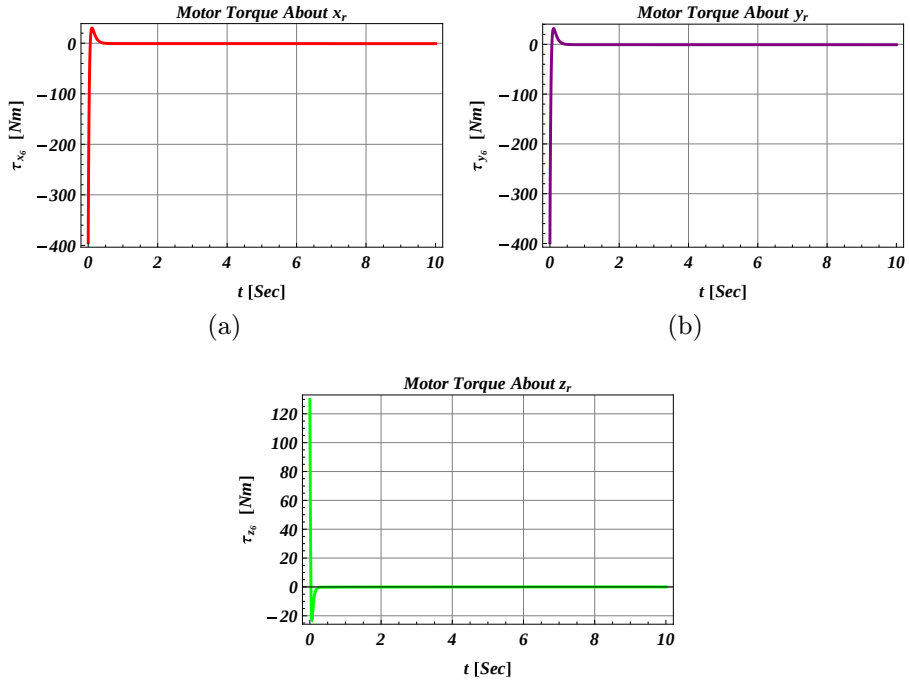


Figure 5.4: Simulation results of the RPY torques, with no position control, for the I.C. (5.1).

5.1.4. Stabilization And Position Control. We now add the position control, as derived in section 4.2. The initial condition are, as in (5.1):

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix}_{(t=0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 15^\circ \\ 15^\circ \\ 0 \end{bmatrix}$$

$$\dot{\theta}_{1(t=0)} = \dot{\theta}_{2(t=0)} = \dot{\theta}_{3(t=0)} = \dot{\theta}_{4(t=0)} = \dot{\theta}_{5(t=0)} = \dot{\theta}_{6(t=0)} = 0$$

We keep in mind that the combination of these angles orient the robot's body in a global angle of about 21° relative to the inertial vertical axis z_0, and that it is considered as the upper boundary limit, where linear approximation can be assumed as good.

Figure 5.7 present the simulation results for the compleat controller. The quick converges of the system to the equilibrium point is clear despite the relatively large initial conditions, detailed in (5.1). It is possible to see the direction and amplitude correlation between the ball's rolls and the robot's RPY angles. As expected, the ball's yaw angle does not converge to 0 but ebb to a constant value, as illuminated in section 4.4.

Figure 5.3g presents the travel of the ball's center of mass on the horizontal ,$x-y$, plane. It is noticeable that although the position P.D. control disturb the partial-exponential stabilization achieved earlier, the robot's body angle is under 5° after only 2 seconds, and maneuver itself towards the desired position.

Figure 5.8 presents the corresponding torques applied by the spherical actuator, in order to achieve the stability, and position control. The high torques and the long travel the pendulum seems to use while stabilizing, are referred at the end of the of subsection 5.1.5.

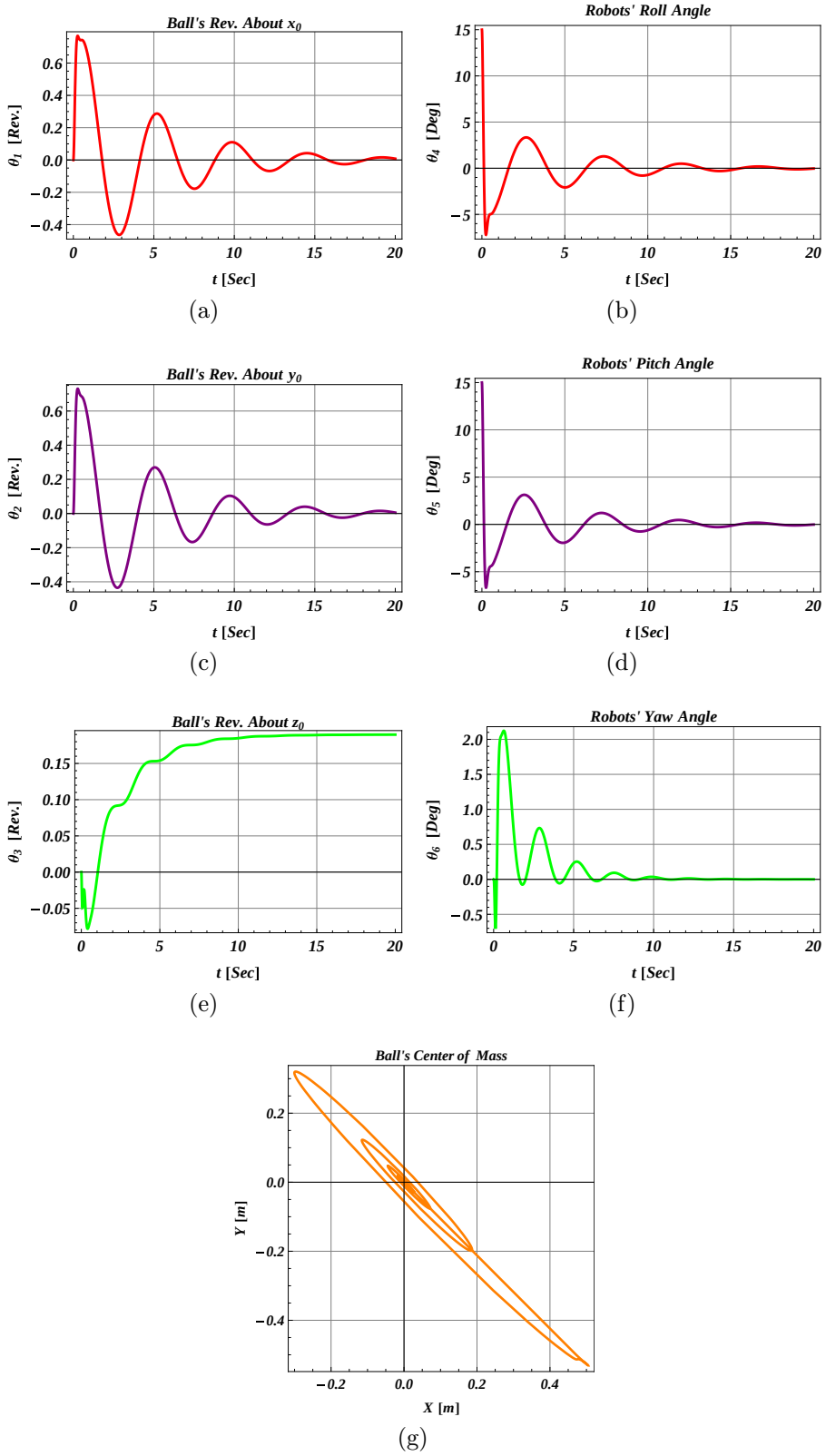


Figure 5.5: Simulation results of the ball's revolutions, the robot's RPY angles and plane travel, considering no added friction, for the I.C. (5.1)

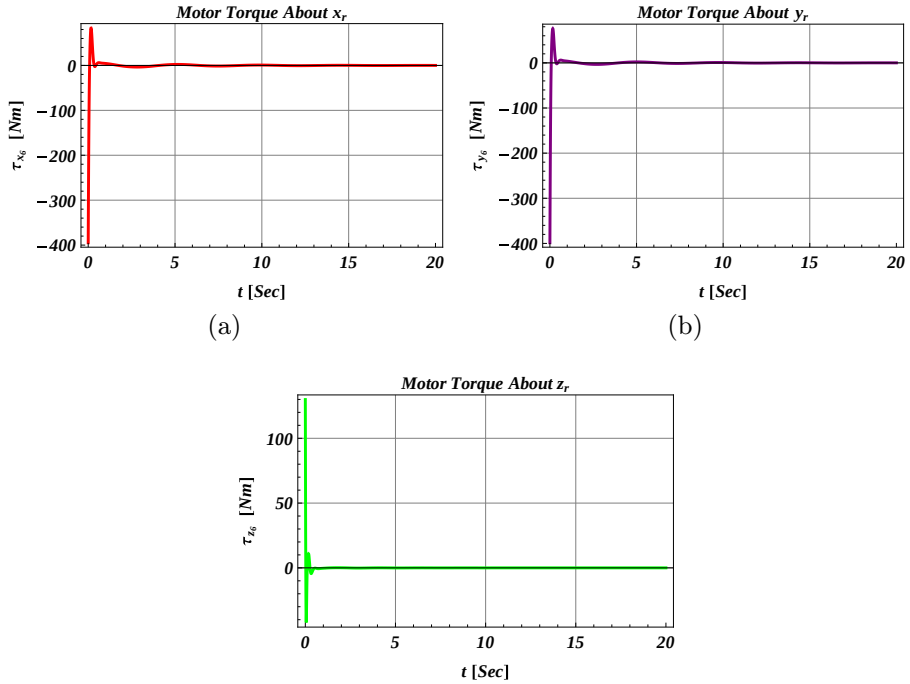


Figure 5.6: Simulation results of the RPY torques, with position control, for the I.C. (5.1).

5.1.5. Stabilization and Position Control under Large Initial Conditions. The overall controller (4.12) is proven as locally stable. As we use the term "local", we wold like to imply that due to the use of linearization, the system's characteristic are approximately kept as long as the system stays in the close neighborhood around the origin. The "local" terms is undetermined and variate between different systems. In the following simulation we would like to push the controller to handel initial conditions that are far off conventional linear approximation boundary. The initial conditions presented in (5.2) yields a global angle of 28° about the inertial vertical axis z_0.

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix}_{(t=0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 20^\circ \\ 20^\circ \\ 10^\circ \end{bmatrix} \tag{5.2}$$

$$\dot{\theta}_{1\,(t=0)} = \dot{\theta}_{2(t=0)} = \dot{\theta}_{3(t=0)} = \dot{\theta}_{4(t=0)} = \dot{\theta}_{5(t=0)} = \dot{\theta}_{6(t=0)} = 0$$

Figure 5.7 present the quick converges of the system to equilibrium point despite the relatively distant initial conditions. It is possible to see the correlation between the ball's and the robot's RPY angles. As expected the ball's yaw angle does not converge to 0 but ebb to a constant value, as it is not controlled. Figure 5.3g shows the travel of the ball on the horizontal ,$x - y$, plane. It is noticeable that the position P.D. control disturb the rapid stabilization achieved earlier.

Figure 5.8 presents the corresponding torques applied in order to achieve stability, and position.

Although the torques and the travel presented in Figures 5.5g and 5.8 may seem large, one should consider the scale of the system. Let us evaluate a simple mathematicl inverted pendulum at a static torque equilibrium. A point mass of a $42[Kg]$ distanced 1 meter off the pivot, Figure 5.9. Further more, the distance of the point mass along the horizontal axis is: $1 \cdot Sin(26^\circ) = 0.43[m]$. These calculations for the simple configuration, puts the sizes scale in the appropriate perspective.

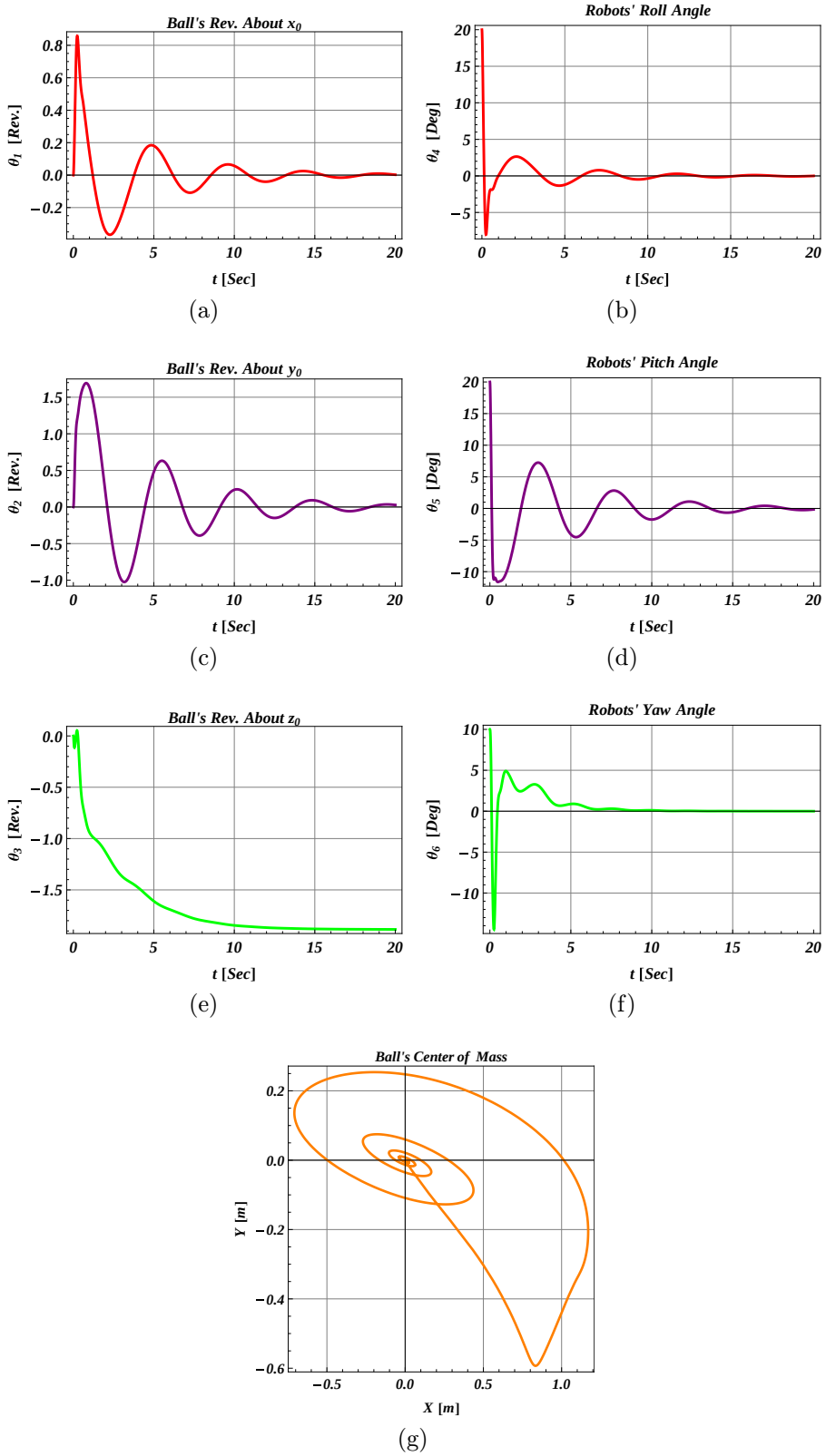


Figure 5.7: Simulation results of the ball's revolutions, the robot's RPY angles and plane travel, considering no added friction, for the I.C. (5.2)

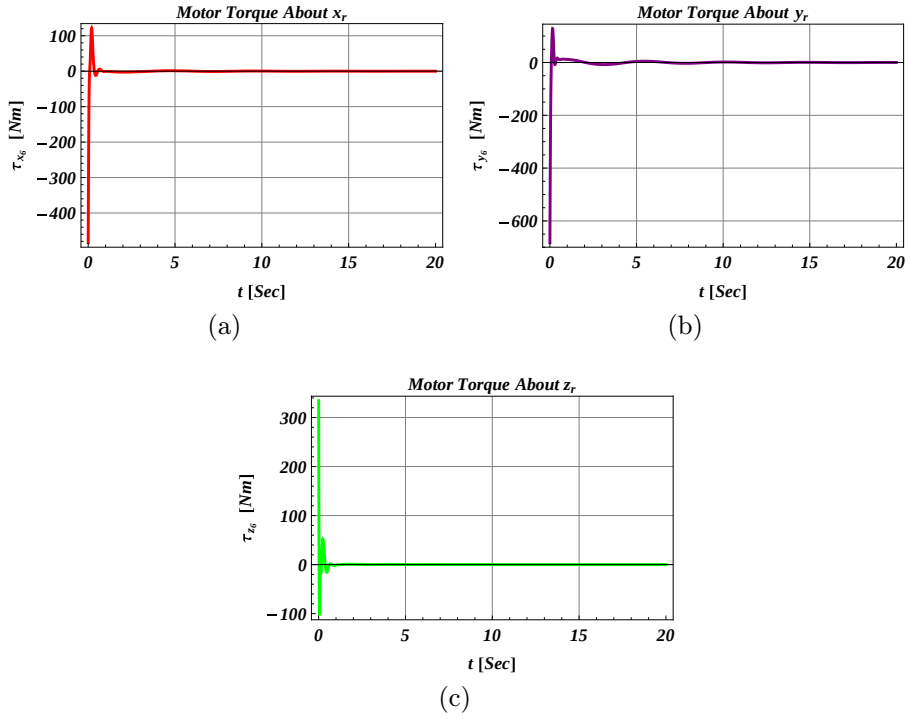


Figure 5.8: Simulation results of the RPY torques, with position control, for the I.C. (5.2).

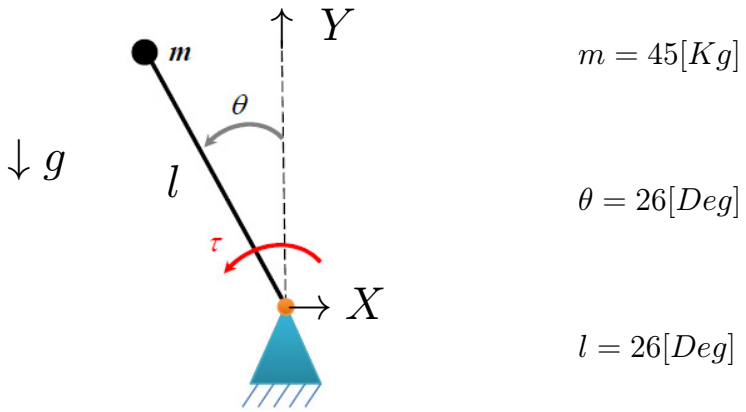


Figure 5.9: A fixed-pivot inverted pendulum with approximate Ballbot parameters oriented following (5.2).

$$\begin{gathered} m \cdot g \cdot l \cdot Sin\left(\theta\right) + \tau = 0 \\ \Downarrow \\ \tau = -45 \cdot 9.81 \cdot 1 \cdot Sin\left((26)^{\circ}\right) = -396.7\left[Nm\right] \end{gathered} \tag{5.3}$$

5.1.6. Trajectory Tracking. After proving the controller capabilities, we shell present its orientation and trajectory following. For the trajectory tracking simulation we use small initial conditions, in order to allow the robot's position controller to be effective, while causing minimal disturbance to the stability controller. Trajectory planning is assembled using several step function combination. The initial conditions are given in (5.4):

$$\begin{gathered} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix}_{(t=0)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1\circ \\ 1^{\circ} \\ 0 \end{bmatrix} \\ \dot{\theta}_{1\,(t=0)} = \dot{\theta}_{2(t=0)} = \dot{\theta}_{3(t=0)} = \dot{\theta}_{4(t=0)} = \dot{\theta}_{5(t=0)} = \dot{\theta}_{6(t=0)} = 0 \end{gathered} \tag{5.4}$$

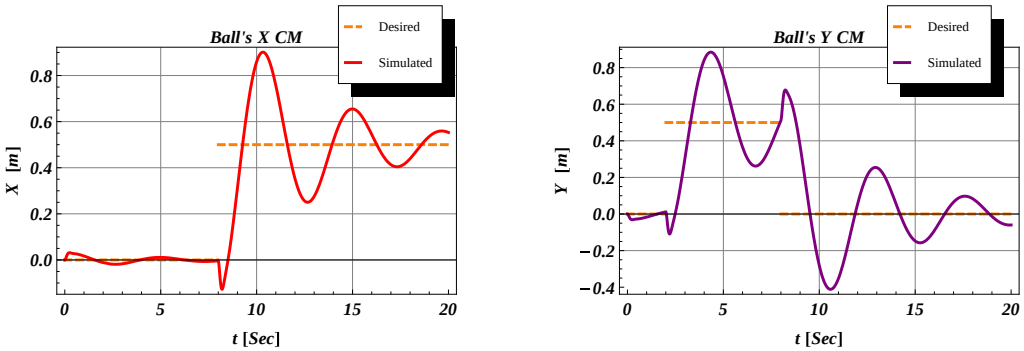
The trajectory sequence presented in Table 5.3 and in the corresponding Figure 5.10. More in Figure 5.10 the simulation results.

Figure 5.10 emphasize the robot's maneuver nature. In order to translate itself to a new location, the robot roles its base to the opposite direction as it is leaning towards the desired direction. It is now travel

Table 5.3: Trajectory Path Plan

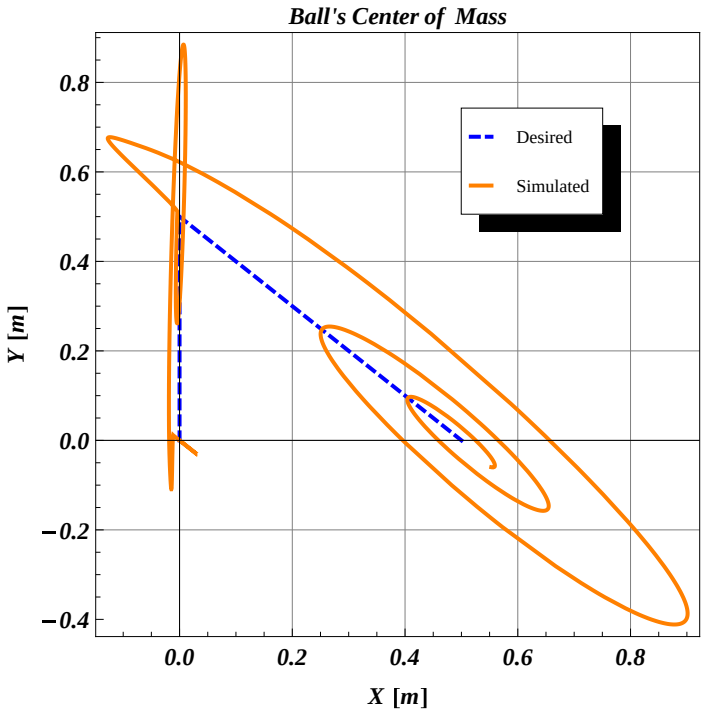
Time Span$[Sec]$	Maneuver
$[0-4]$	stabilizing
$[4-10]$	Travel to $(0,+0.5)\,[m]$.
$[10-20]$	Travel to $(+0.5,0)\,[m]$, diagonal path.

along the desired trajectory with a slight angle off the vertical in the direction of progress. When arriving near the desired location, the stabilization control is "taking the lead" as the ball roles faster then the body's linear velocity in order to lean the robot's body backwards, and by that returning to the vertical position.



(a)

(b)



(c)

Figure 5.10: Step based designed sequence for the robot's: (a) X position, (b) Y position and (c) the simulation result.

5.1.7. Varying Trajectory Tracking. In order to present a varying trajectory tracking, the same small initial conditions as (5.4) are used.

The trajectory is sinusoidal along the y_0 axes and has a constant velocity along the x_0 axes. In Figures 5.11a and 5.11b the desired trajectory is presented. Notice that the trajectory initializes only after 4 seconds, in order to enable the stabilization controller to stabilize the robot before following the path. This simulation is presented along time span of $30[sec]$.

The close tracking of the ball's center of mass to the desired trajectory is obvious in Figure 5.11e. It is noticeable in Figures 5.11c and 5.11e that the trajectory initializing before the robot is fully stabilized, so the tracking is in transient stage.

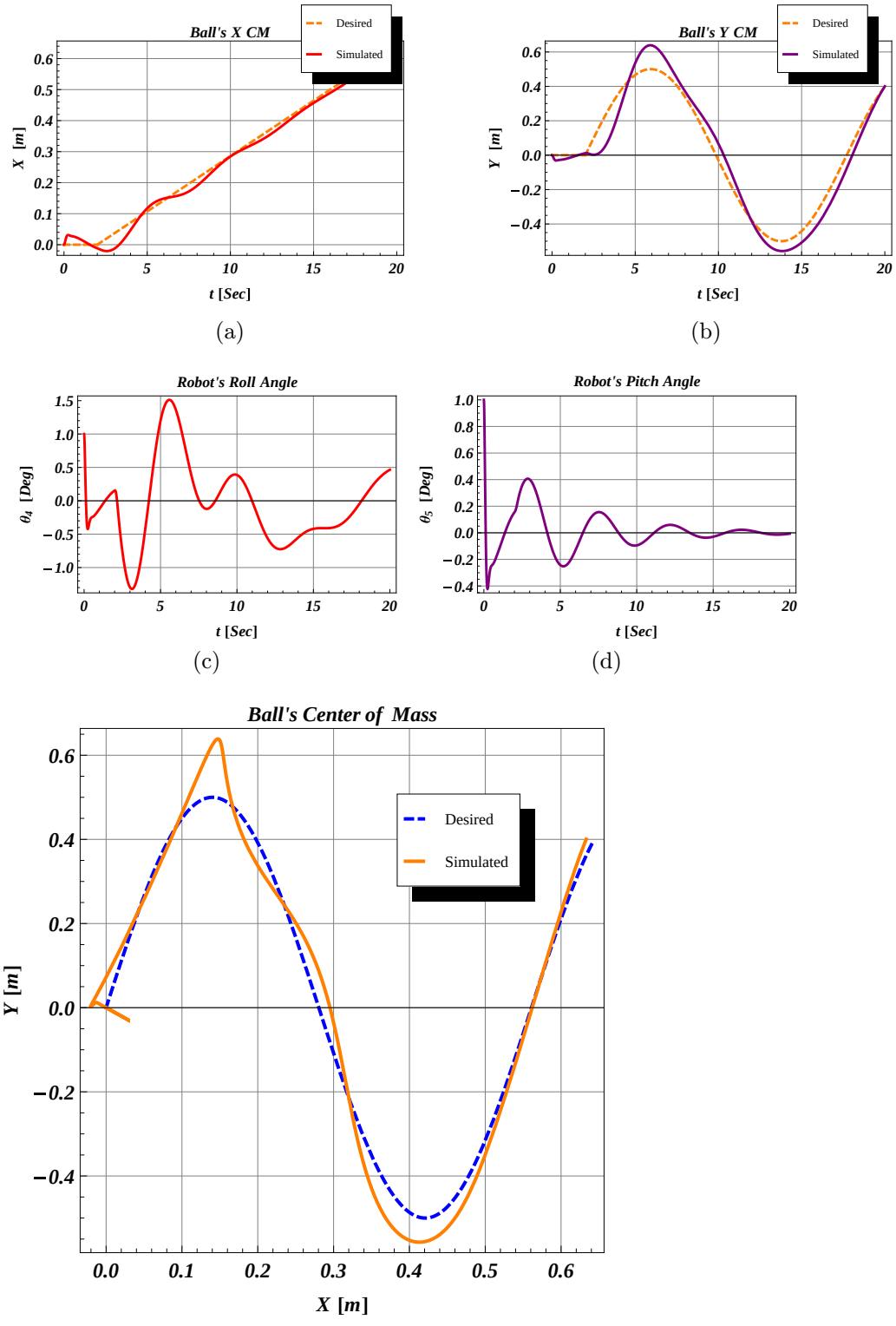


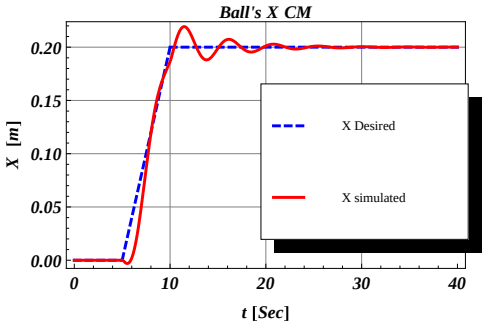
Figure 5.11: Sinusoidal designed trajectory for the robot, and results: (a) x position, (b) y position, (c) body's Roll angle (d) body's Pitch angle (e) trajectory tracing.

5.1.8. Results Comparison with C.M.U Simulations. In this subsection we try to emulate the simulations done by T.B.Lauwers, G.A Kantor and R.L.Hollis in [**3, 39**]. Although we do not have a full size experimental system, we would like to show improvement in the system performance using the current controller. All of the trajectories tracking starts with zero state initial conditions, $x = [q^T, \dot{q}^T]^T = 0$, and simulated along time span of $40[sec]$.

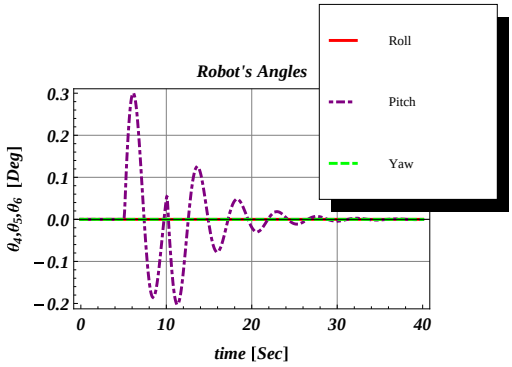
Two Simulation are performed: step response and square trajectory. The step response is assembled of a $0.2[m]$ offset along the x axis, smoothed by a linear path of $\frac{1}{25}\left[\frac{m}{sec}\right]$ for $5[sec]$, while the system is station keeping along y axis, Figure 5.12a. The close tracking and the fast converges of the system is evident in Figure 5.12a. In Figure 5.12b the robot's minor RPY angles along the trajectory tracking are presented. As one can see, the largest angle is only 0.3° and rapidly converge .

The square trajectory is assembled of four points, each positioned as a target in a $10[sec]$ interval, Figure 5.12f. Figures 5.12c ,5.12d presents the position error in x, y direction and 5.12e presents RPY angles of the robot along the trajectory tracking.

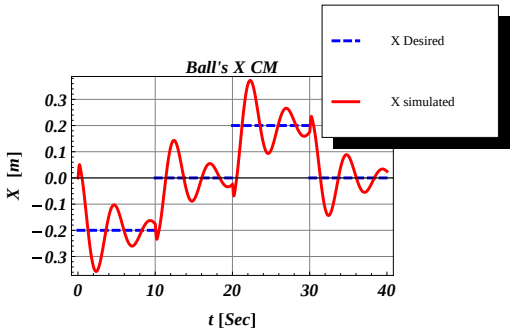
When comparing this results to [**3, 39**], we notice the major improvement in performance in the system's stabilization, as the RPY angles converge quickly. The position control present just a slight improvement in performance, while the position converges along with the body's angles.



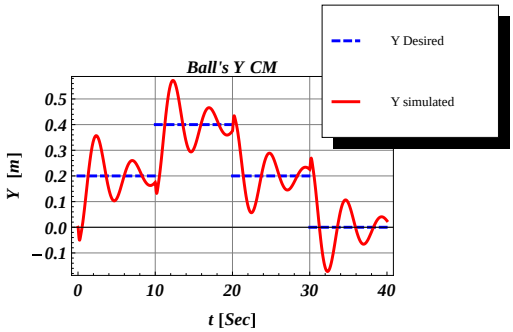
(a)



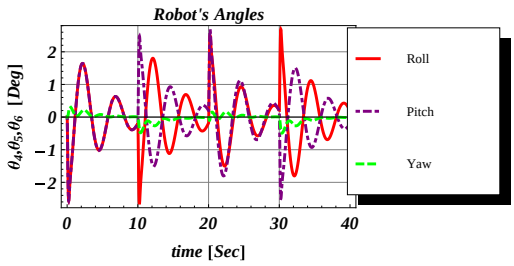
(b)



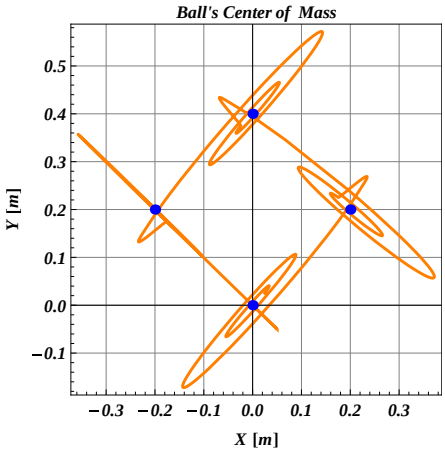
(c)



(d)



(e)



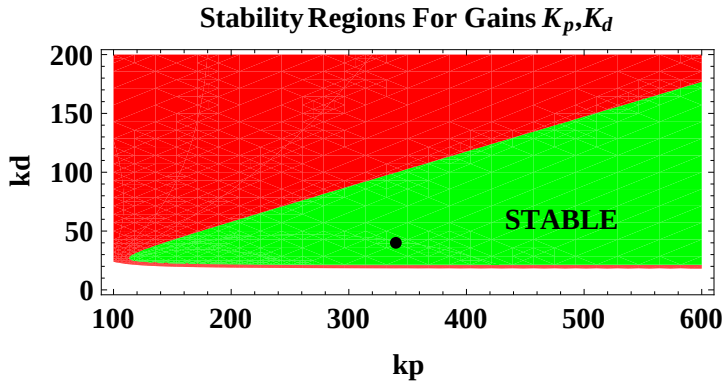
(f)

Figure 5.12: Compared results of CMU Ballbot's experiments [**3, 39**], using the current system. (a,b): step response, (c,d,e,f): Square trajectory

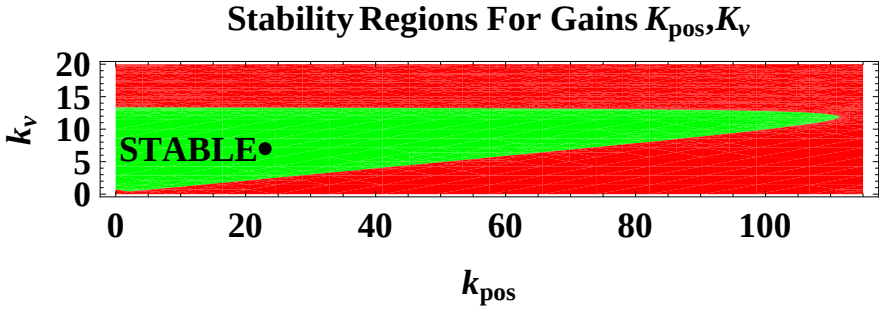
5.1.9. Effect of Controller's Gains on stability. In the next subsection we present a visual mapping of the system stability as dependent in the controller's gains. The stability of a system with given dimensions, depends on many variables such as the four controller gain K_p, K_d, K_{pos}, K_v, where each one of them is a three elements diagonal matrix - for the simple case. That means we are dealing with a minimum of twelve parameters independently effecting the closed loop system behavior. Although full parametric investigation of the system is out of the scope of the current work, we present its tip of the iceberg, in order to achieve some intuition about the relations between the controller's gains to the system's stability.

Stability investigation is based on the calculation of the real part of the eigenvalues of the matrix A, as described in Subsections 4.2 and 4.3. Four binary plots are presented, where each plot describe the system's stability by the definition of the eigenvalues criteria, as a function of two variates parameters. For each plot we take all of the other parameters as presented in Tables 5.1 and 5.2. Further more, because each controller gain is a three element diagonal matrix, we keep the ratio between the gain matrix elements as presented in Table 5.2, and variate a multiplier to the matrix in some range. One can observe that the parameters presented in Table 5.2 is contained in the "stable" area of the plots, marked with a black dot.

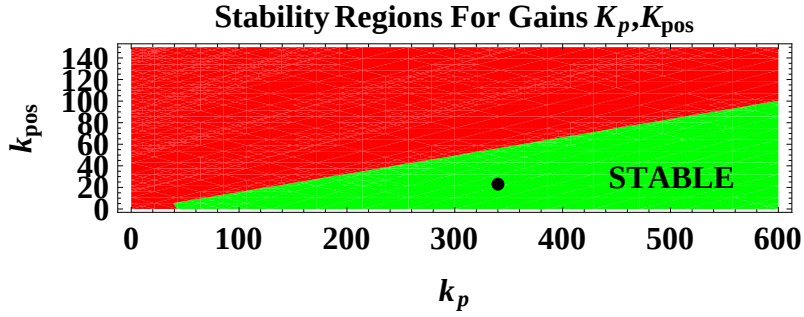
we can see that the proportional stabilization gain K_p has a minimum which under it, it is ineffective, and on the other hand as it get bigger it allow wider variety of other gains Figures 5.13a and 5.13c.



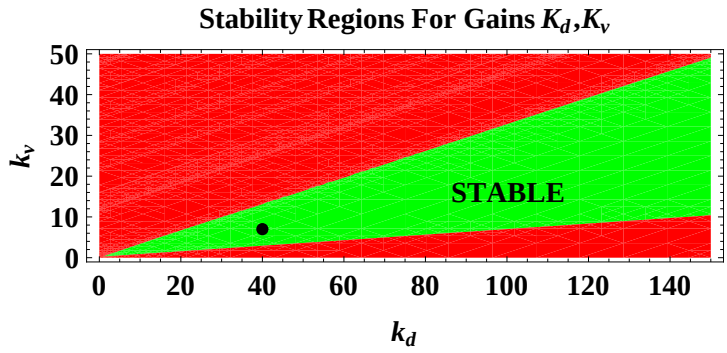
(a)



(b)



(c)



(d)

Figure 5.13: The effect of the change in the controller gains to the system's stability. The changes are in the gains:
(a) K_p, K_d (b) K_{pos}, K_v (c) K_p, K_{pos} (d) K_d, K_v.

It is seen in figures 5.13a, 5.13d and 5.13b that the system is much less tolerant to the changes in the gains proportional to the parameters velocity K_d and K_v.

Of course, as we use bigger gains we increase the control effort, which are the torques applied by the motors. Due to physical constraints it may not be possible to achieve this torques, in a build model. In that case, an optimization can be done in order to achieve maximum performance with minimal control effort. The current controller gains were tuned manually, and as can be seen in Figures 5.13a and 5.13b, they are fairly close to the minimal amplification edge.

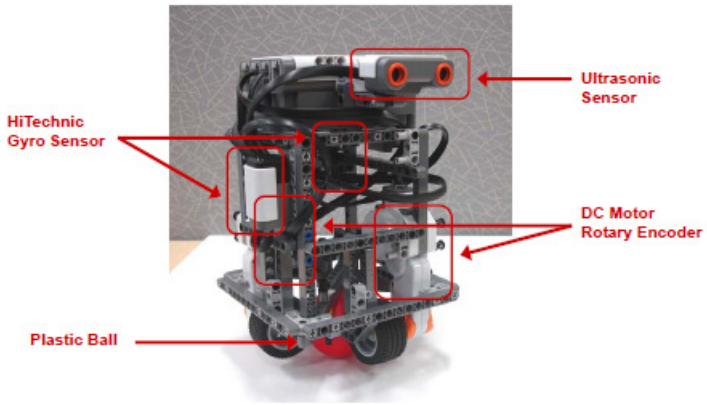


Figure 5.14: Experimental Ballbot system - LEGO NXT.

5.2. Experiments

We now review some of the important subjects that has to be considered while preforming experiments, and were not considered at the theoretical and simulation chapters.

5.2.1. Experimental System. In order to demonstrate the controller performances, a reduced Ballbot system was built using the LEGO NXT MindStorm kit. The robot's main body and the relevant software, are based on the work of Yorihisa Yamamoto [41]. The basic model is a boxed structure holding the NXT programmable "brick" at its top, two motors for Roll and Pith angles and two corresponding gyroscopes. An ultrasonic distance sensor is mounted as well, and will not be discussed here - Figure 5.14. In order to simulate the desired control law, a designated MATLAB Simulink Toolbox is used, who's capable of compile and upload the software to the NXT Brick. A special compiler is needed as described in [41].

The original NXT Ballbot's control law used the LQR method, where the linear system was achieved by preforming linearization twice. One where the assumption that the two direction, Roll and Pitch, are decoupled, so the robot can be referred as two separate unicycles, and totaly discard the yaw angle. Second, each planer unicycle's equation of motion were linearized about its upper equilibrium point.

The Dynamics and control law derived in Chapters 3 and 4, are nonlinear and obtain the addition Yaw rotations of the ball and of the robot. Therefore, the control law had to be reduced in order to fit the to the actual system. First, all of the elements in yaw direction where assumed as 0, *i.e.* $\{\theta_3, \theta_6, \dot{\theta}_3, \dot{\theta}_6, \tau_y\} = 0$. Next, due to the NXT brick computation capabilities, the control law coefficient matrixes D_4, s_4, had to be linearized about the upper equilibrium point. Eventually the control gains K_p, K_d, K_{pos}, K_v, were tuned in a tray and error procedure. All of the systems parameters, such as lengthes, masses, moments of inertia, are driven out of [**41**]. the parameters and the controller gains detailed in table 5.4.

Simulated results for the LEGO NXT system parameters and controller gains, as introduced in Table 5.4, are presented in Figure 5.15. The following Figure 5.16, presents the estimated motor torques. For the simulation we assume the following initial conditions:

$$\{\dot{\theta}_i = 0\,;\, i = 1 \ldots 6 \quad , \quad \theta_4 = \theta_5 = 5^\circ \quad , \quad \theta_i = 0\,;\, i = 1, 2, 3, 6\} \quad (5.5)$$

The experimental system was measured using a spatial Infrared cameras, connected together in order to calculate the absolute position

Table 5.4: Experimental System's and Controller Parameters

Ball's radius	r	$=$	$0.026\ [m]$
Ball's mass	m_b	$=$	$0.013\ [Kg]$
Robot's mass	m_r	$=$	$0.682\ [Kg]$
Distance of the robot's cm from the ball's cm	l	$=$	$0.17\ [m]$
Ball's Tensor of Inertia	I_b	$=$	$\text{Diag}(\frac{2}{3}mbr^2)\ [Kg \cdot m^2]$
Robot's Tensor of Inertia	$\tilde{I}_r$	$=$	$\text{Diag}(\frac{1}{3}mrl^2, \frac{1}{3}mrl^2, \frac{1}{2}mr0.076^2)\ [Kg \cdot m^2]$
Viscous Friction Coefficients	$fric$	$=$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.002 & 0 & 0 & 0.002 & 0 & 0 \\ 0 & -0.002 & 0 & 0 & 0.002 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \left[\frac{Kg \cdot m^2}{sec}\right]$
Feedback Linearization - Proportional gains	K_p	$=$	$\begin{bmatrix} 1200 & 0 & 0 \\ 0 & 1200 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Feedback Linearization - Differential gains	K_d	$=$	$\begin{bmatrix} 255 & 0 & 0 \\ 0 & 255 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Position Control - Proportional gains	K_{pos}	$=$	$\begin{bmatrix} 0.015 & 0 & 0 \\ 0 & 0.0157 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Position Control - Differential gains	K_{vel}	$=$	$\begin{bmatrix} 0.027 & 0 & 0 \\ 0 & 0.027 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

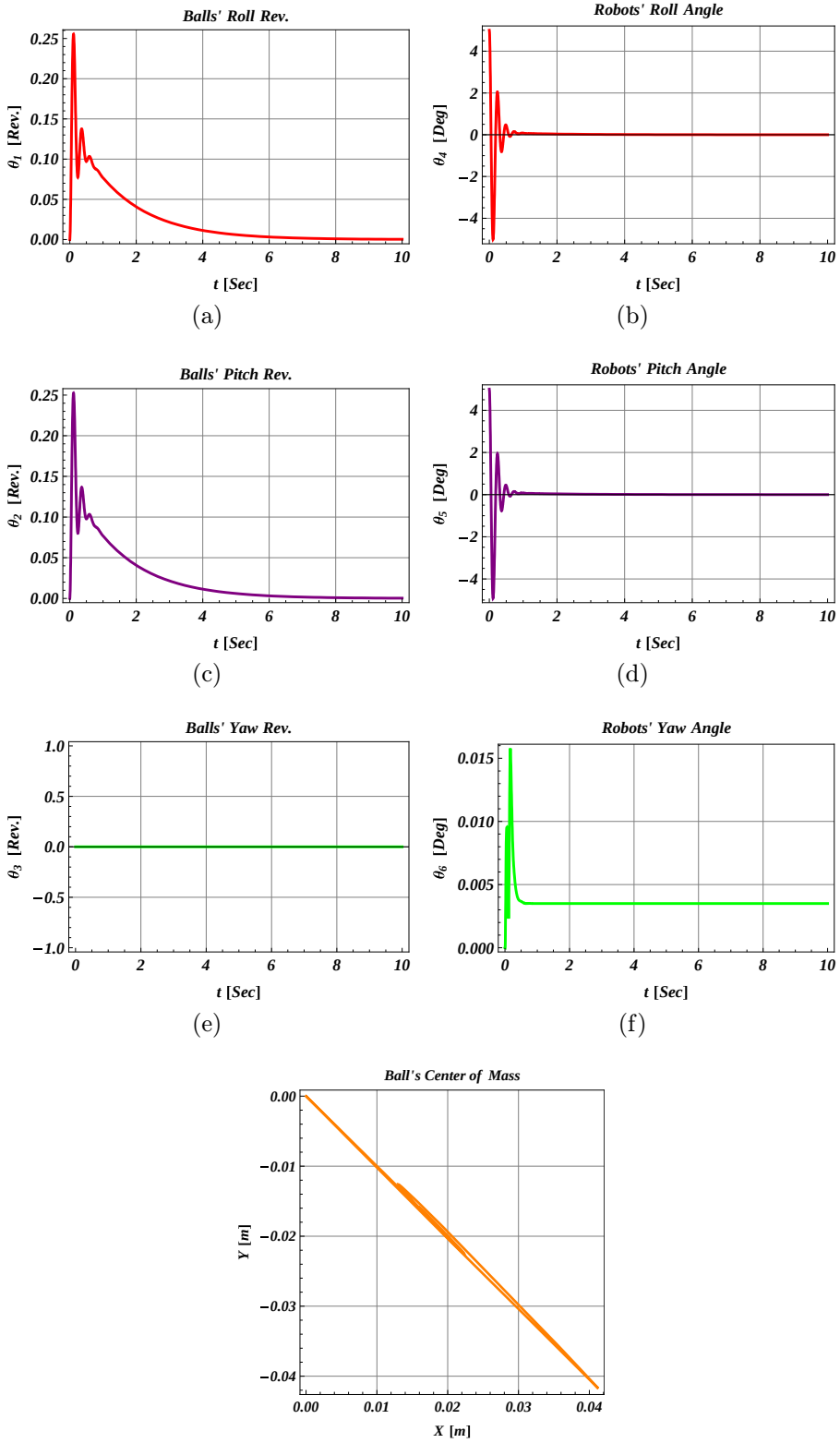


Figure 5.15: Simulation results of the ball and robot's RPY angles and plane travel of the LEGO NXT experimental system, under the I.C. 5.5

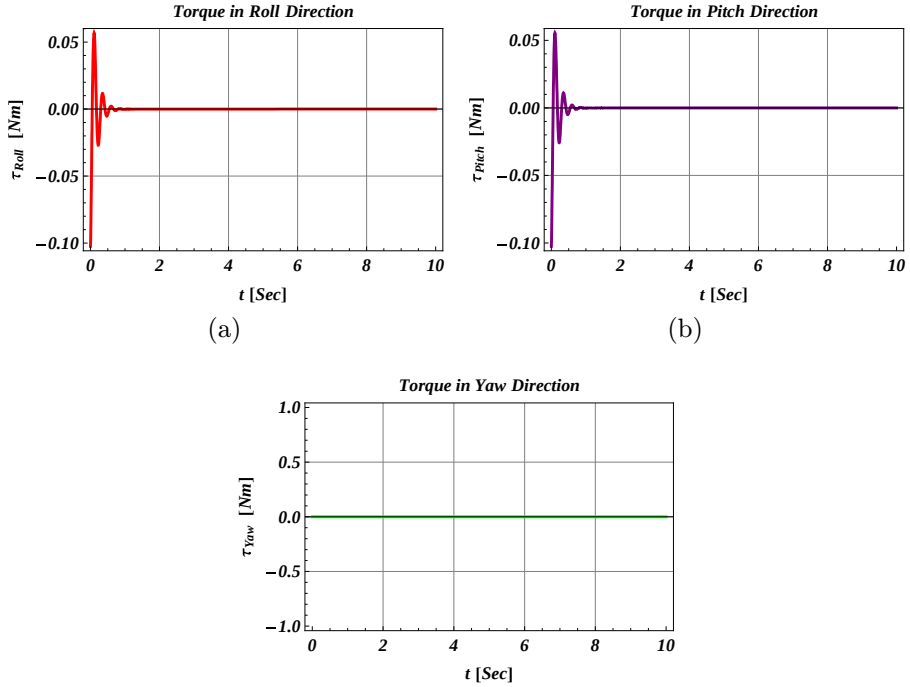
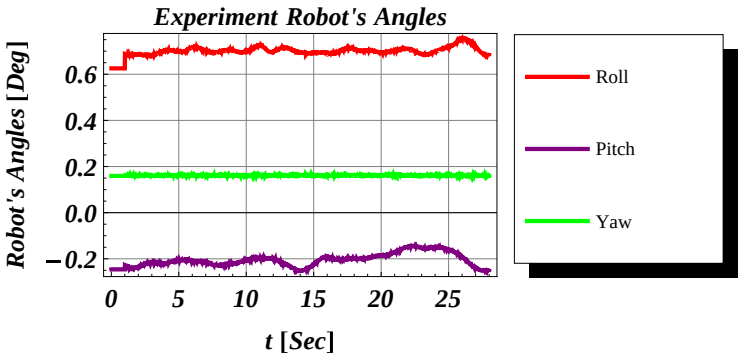
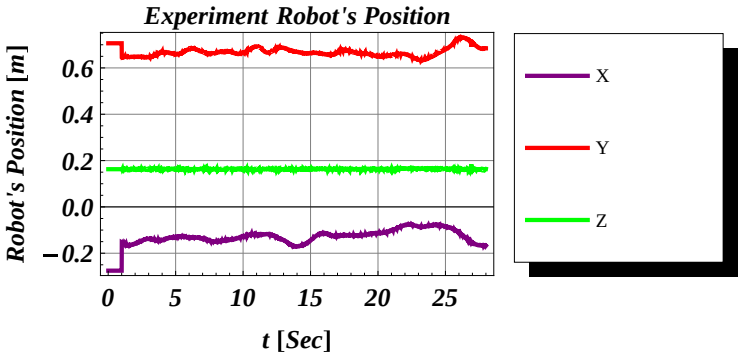


Figure 5.16: Simulation results of the RPY torques for the LEGO NXT experimental system, under the I.C. 5.5.

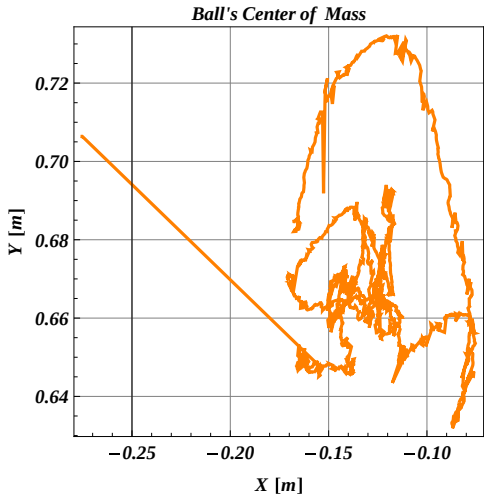
and orientation of the body. The results are presented in Figure 5.17. the correlation between the RPY angles and the X,Y,Z position are clear in Figures 5.17a and 5.17b. The travel of the robot in Figure 5.17c shows the needed travel in order to stabilize the robot, and the position keeping after stabilization. We note that the LEGO ballbot system is not utterly symmetrical, so the system is in its equilibrium point when the robot's body is slightly tilted, as seen in Figure 5.17a.



(a)



(b)



(c)

Figure 5.17: Experiment results of the robot's RPY angles and plane travel of the LEGO NXT experimental system

CHAPTER 6

Conclusions and Future Work

6.1. Conclusions

In this work we have derived and studied the full 3D dynamic model of the underactuated Ballbot system. The holonomic nature of the motion of the ball rolling on a plane was discussed, and a new approach considering the motion of the center of the ball as holonomic, was described and implemented to the kinematic model of the system. a Non-linear control law was constructed in two parts, stabilization control and position control. The stabilization control was derived using the partial feedback linearization method, and was proven to exponentially stabilize the robot's orientation only. When including a PD position control law, the overall controller prove the system to be locally stable by separating the passive unstable part of the system and by using Lemma 4.2. Conditions for the right selection of the controller gains are presented, and the correlation between the controller gains to the system's behaviour is referred. The contribution of the additional embedded viscous friction to the system's stability was reviewed and explained. The derived spatial model is more advanced then its previous, as it consider all of the DoF of the rigid bodies. More, although the linearized model of this system shows independency of the DoF ,

it is still consider the two rotations of the ball and the body about z_0 as DoF , which are not considered in earlier models.

Simulation were preformed in order to demonstrate the controller's capabilities and characteristic behaviour. It is further shown that although the controller prove the system as locally stable, it can handle relatively large initial conditions. The conditions of angles larger then 20°, are usually referred as out of the boundaries, where linearization is considered a good approximation. This characteristics of the controller provide a very large converges basis of attraction. Following C.M.U. Ballbot's simulations, presenting simulated converges of about $20[sec]$ for $0.2[m]$ step trajectory while the system further oscillate in 0.6°, we provide a better trajectory tracking abilities, and faster full converges of the system.

Experimental reduced system were build using the LEGO NXT Mind-Storm kit, assisted by a designated MATLAB-simulink© environment.

6.2. Future Work

In future work we intend to calculate the basin of attraction of the system. We plane to preform an optimization to the systems parameters, in order to find the best ratio between the ball's and the body's dimensions, that will provide maximum stability and minimal control effort. Building a full sized, human hight ballbot, and a Ballbot based

transportation platform are in develop. Another analytical and experimental study about the system behaviour on a non horizontal terrain is one of the future study options. The Yaw rotation of the robot's body is one of the system's capabilities, it's influence to the system can be more comprehensively studied and tuned. Parametric investigation of the influence of the controller gains to the system behavior, as presented in Subsection 5.1.9 can be further expanded.

Bibliography

[1] Independence ibot 4000 mobility system,http://www.ibotnow.com.

[2] Segway personal transporter,www.segway.com.

[3] *A Dynamically Stable Single-Wheeled Mobile Robot with Inverse Mouse-Ball Drive*, Pennsylvania, USA.,, May 2006. IEEE Int. Conf. on Robotics and Automation,.

[4] D.J. Achenson. A pendulum theorem. *Mathematical and physycal Sciences*, 443:239–245, 1993.

[5] D.J. Acheson. Multiple-nodding oscillations of a driven inverted pendulum. *Proceedings Mthematical and Physical sciences*, 448:89–95, 1995.

[6] Domenico Prattichizzo Andrea Gorelli Antonio Bicchi, Andrea Balluchi. Introducing the "spericle": an experimental testbed for research and teaching in nonholonomy. In *icra IEEE Int. Conf. on Robotics and Automation*, pages 2620–2625, Albuquerque, New Mexico, 1997.

[7] A.M. Bloch, PS Krishnaprasad, J.E. Marsden, and R.M. Murray. Nonholonomic mechanical systems with symmetry. *Archive for Rational Mechanics and Analysis*, 136(1):21–99, 1996.

[8] A.V. Borisov and I.S. Mamaev. Rolling of a rigid body on plane and sphere. Hierarchy of dynamics. *Regular and Chaotic Dynamics*, 7(2):177–200, 2002.

[9] B. Browning, J. Searock, P.E. Rybski, and M. Veloso. Turning Segways into Soccer Robots.

[10] Eugene I. Butikov. On the dynamic stabilization of an inverted pendulum. *American Journal of Physics*, 69(7):755–768, 2001.

[11] Antonio Bicchi Carlo Camicia, Fabio Conticelli. Nonholonomic kinematics and dynamics of the sphericle. In *Intelligent Robots and Systems*, volume 1, pages 805–8–10, Takamatsu, Japan, November 2000.

[12] A.M. Formals'kii. An inverted pendulum on a fixed and a moving base. *Applied Mathematics and Mechanics*, 70(1):56–64, 2006.

[13] R. Grimshaw. *Nonlinear ordinary differential equations*. CRC, 1993.

[14] John L. Safko H.Goldstein, Charles P. Poole. *Classical Mechanics 3rd Edition*. Addison Wesly, 2001.

[15] M.W. Hirsch and S. Smale. *Differential equations, dynamical systems, and linear algebra*. Academic Press Inc, 1974.

[16] R.G. LANGLOIS J.B. HOLLAND, M.J.D. HAYES. A slip model for the spherical actuation of the atlas motion platform. *TRANSACTIONS- CANADIAN SOCIETY FOR MECHANICAL ENGINEERING*, 29(05-CSME-44):711–720, 2005.

[17] B.D. Johnson. The nonholonomy of the rolling sphere. *American Mathematical Monthly*, 114(6):500–508, 2007.

[18] P.L. Kapitza. *Dynamic stability of the pendulum with vibrating suspention point*, volume 21. Nauka Scientific-Moscow, 1951. (in Russian).

[19] H.K. Khalil. *Nonlinear systems*. Pearson Education, third edition.

[20] Gabor Kusha. *Dynamics and Control of a ManeuveringInverted Pendulum on an Uneven Surface*. PhD thesis, The Technion-ISRAEL Institute of Technology, 2001.

[21] T. Lauwers, G. Kantor, and R. Hollis. One is Enough! In *Proc. Intl. Symp. for Robotics Research*, pages 12–15. Springer.

[22] ROBERT G. LANGLOIS ABRAHAM WEISS M. JOHN, D. HAYES. Atlas motion platform generalized kinematic model. In M. Gouttefarde S. Krut N. Andreff, O. Company and F. Pierrot, editors, *Proceedings of the Second International Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, Montpellier, France, September 2122, 2008.

[23] Takaya Ochiai Masaaki Kumagai. Development of a robot balancing on a ball. In *ICCA 2008 Proceedings*, pages 433–438, Oct 2008.

[24] M. Montemerlo and S. Thrun. Large-scale robotic 3-d mapping of urban structures. In *Proc. of the Int. Symp. on Experimetnal Robtoics (ISER)*. Springer, 2004.

[25] R.M. Murray, Z. Li, and S.S. Sastry. *A mathematical introduction to robotic manipulation*. CRC, 1994.

[26] M.Vidyasagar M.W.Spong. *Robot Dynamics And Control*. John Wiley &Sons,Inc., 1989.

[27] R. Nakajima, T. Tsubouchi, S. Yuta, and E. Koyanagi. A development of a new mechanism of an autonomous unicycle. In *Intelligent Robots and Systems, 1997. IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on*, volume 2, 1997.

[28] H.G. Nguyen, J. Morrell, K. Mullens, A. Burmeister, S. Miles, N. Farrington, K. Thomas, and D. Gage. Segway robotic mobility platform. *SPIE Mobile Robots XVII*, 2004.

[29] G. Oriolo and Y. Nakamura. Control of mechanical systems with second-order nonholonomic constraints: Underactuated manipulators. In *Conference on Decision and Control*, pages 2398–2403. Citeseer, 1991.

[30] R. Ortega, MW Spong, F. Gomez-Estern, and G. Blankenstein. Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment. *IEEE Transactions on Automatic Control*, 47(8):1218–1233, 2002.

[31] R. Ortega, MW Spong, F. Gomez-Estern, and G. Blankenstein. Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment. *IEEE Transactions on Automatic Control*, 47(8):1218–1233, 2002.

[32] Alan Parekh. Hacked gadgets,http://hackedgadgets.com/2006/08/11/ballbot-balances-itself-on-one-ball/, 8 2006.

[33] WEI Qifeng, W.P. Dayawansa, and W.S. Levine. Nonlinear controller for an inverted pendulum having restricted travel. *Automatica*, 31(6):841–850, 1995.

[34] A. Shapiro, E. Rimon, and JW Burdick. On the mechanics of natural compliance in frictional contacts and its effect on grasp stiffness and stability. In *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04*, volume 2, 2004.

[35] M.W. Spong. The control of underactuated mechanical systems. In *First international conference on mechatronics*, pages 26–29, 1996.

[36] M.W. Spong. Underactuated mechanical systems. *Lecture Notes in Control and Information Sciences*, pages 135–150, 1998.

[37] George Kantor Tom Lauwers and Ralph Hollis. *Robotics Research Results of the 12th International Symposium ISRR*, volume 28/2007 of *Springer Tracts in Advanced Robotics*. Springer Berlin / Heidelberg, 2007. One is Enough!

[38] George A. Kantor Umashankar Nagarajan, Anish Mampetta and Ralph L. Hollis. State transition, balancing, station keeping, and yaw control for a dynamically stable single spherical wheel mobile robot. *Proceedings of the IEEE International Conference on Robotics & Automation*, pages 998–1003, May 2009.

[39] Ralph L. Hollis Umashankar Nagarajan, George Kantor. Trajectory planning and control of an underactuated dynamically stable single spherical wheeled mobile robot. IEEE International Conference on Robotics and Automation, May 2009.

[40] H. Uustal and J.L. Minkel. Study of the Independence IBOT 3000 Mobility System: an innovative power mobility device, during use in community environments. *Archives of physical medicine and rehabilitation*, 85(12):2002–2010, 2004.

[41] Yorihisa Yamamoto. Nxt ballbot control design and building instructions,http://www.mathworks.com/matlabcentral/fileexchange/23931, Apr 2009.

[42] C. Ye and J. Borenstein. Obstacle avoidance for the segway robotic mobility platform. In *ANS 10th Int. Conf. on Robotics and Remote Systems for Hazardous Environments*, pages 107–114, 2004.

Appendices

APPENDIX A

Exponential Converges Lemma

Following [15, p.180] , we present a proof to the dependency of the exponential converges of a nonlinear system, in the properties of its linear approximation at a close neighborhood of its equilibrium point.

LEMMA A.1. *Let a nonlinear dynamical system be of the form*

$$\dot{x} = f(x); \quad f : W \to \mathbb{R}^n; \quad W \subset \mathbb{R}^n \text{ open}, \tag{A.1}$$

Where $\bar{x} \in W$ is an equilibrium point, so that $f(\bar{x}) = 0$. Let Ax be a linear approximation of $f(x)$, so that

$$\begin{aligned} \dot{x} &\approx Ax, \\ A &= \left.\frac{\partial f(x)}{\partial x}\right|_{x=\bar{x}}. \end{aligned} \tag{A.2}$$

Let $\lambda_i(A)$ be the i^{th} eigenvalue of A $i = \{1, \ldots, n\}$, and notate the initial conditions $x(t = 0) = x_0$.

If all $Re(\lambda_i(A)) < -c$ for $0 < c \in \mathbb{R}$, then there exist a neighborhood $U \subset W$ of $\bar{x}$ such that

(a) $x(t) \in U \quad \forall x_0 \in U$, while $\quad 0 < t$.

(b) For any norm on $\mathbb{R}^n$ there is a constant $0 < B$ such that

$$|x(t) - \bar{x}| \leq B \cdot e^{-t \cdot c} \, |x_0 - \bar{x}| \, ; \quad \forall x_0 \in U, \quad 0 \leq t.$$

PROOF. For the convenience, and without the lost of generality, we may choose $\bar{x} = 0$.

Denote b such that $0 < c < b \in I\!R$, such that all of the eigenvalues of $Re(\lambda_i(A)) < -b$. As in [15] and as known from linear algebra

$$\langle Ax, x\rangle \leq -b\,|x|^2\,, \tag{A.3}$$

Due to the definition of the equilibrium point $f(0) = 0$ and the linear approximation A of $f(x)$, by following the derivative definition we write

$$\lim_{x\to 0}\left(\frac{|f(x) - Ax|}{|x|}\right) = 0.$$

Multiply the inner function by $\frac{x}{x}$ and applying Cauchy's inequality $|a \cdot b| \leq |a| \cdot |b|$, yields

$$\lim_{x\to 0}\left(\frac{\langle f(x) - Ax, x\rangle}{|x|^2}\right) = 0. \tag{A.4}$$

Now, as

$$\lim_{x\to 0}\left(\frac{\langle f(x) - Ax, x\rangle}{|x|^2}\right) = \lim_{x\to 0}\left(\frac{\langle f(x), x\rangle - \langle Ax, x\rangle}{|x|^2}\right),$$

we substituting (A.3) and obtain

$$\lim_{x\to 0}\left(\frac{\langle f(x), x\rangle + c\,|x|^2}{|x|^2}\right) \leq 0,$$

So there exist $0 < \delta$ small enough that if $|x_0| \leq \delta$, therefore $x(t) \in U$, where $U \subset W$ is the basin of attraction of $x(t)$, and it is stem that

$$\langle f(x), x\rangle \leq -c\,|x|^2\,. \tag{A.5}$$

Now, because

$$\frac{\partial}{\partial t}|x| = \frac{\partial}{\partial t}\left(\frac{x^T \cdot x}{|x|}\right) = \frac{\langle \dot{x}, x\rangle}{|x|} = \frac{\langle f(x), x\rangle}{|x|},$$

while considering (A.5) we obtain

$$\frac{\partial}{\partial t}|x| \leq -c\,|x|\,. \tag{A.6}$$

Equation (A.6) implies that $|x(t)|$ is decreasing, so $x(t) \in U$ because $(x_0, \bar{x}) \in U$. Moreover it implies that

$$|x(t)| \leq e^{-c\cdot t} \cdot |x_0| \quad \forall 0 \leq t.$$

So there is a neighborhood U of $\bar{x}$, small enough that if $x_0 \in U$, the nonlinear dynamical system converge exponentially to $\bar{x}$ like its linear approximation. □

תקציר

מטרת עבודה זו היא הקמה וחקירה של בקרה לא לינארית עבור מטוטלת הפוכה מרחבית. עיקר המוטיבציה הוא שיפור המודל הדינאמי וחוק הבקרה למערכת "Ballbot". המערכת הינה חד-אופן תלת מימדי. המודל מבוסס על סוג של מטוטלת הפוכה ולכן, בעבודה זו נסקרים מספר סוגי מטוטלות הפוכות ומאפינן. בנוסף מפותח מודל מרחבי עבור המערכת הדינאמית, ומוקם בקר המתבסס על שיטת "Partial Feedback-Linearization" של מערכות תת – ממונעות. מוצגות סימולציות של המערכת עבור מספר תנאי התחלה, יציבות, עקיבה אחר מסלול ועבור מספר מאפיינים של הבקר. בנוסף, מוצגת חקירה פרמטרית ראשונית של חוק הבקרה. המערכת מוכחת כיציבה על סמך קירוב לינארי של החוג הסגור, תוך מתן קריטריון ליציבות התלוי בהגברי המשוב. על אף הוכחת היציבות תחת קירוב לינארי, המערכת מציגה ביצועים מרשימים עבור תנאי התחלה מרוחקים מנקודת שיווי המשקל. הוקמה מערכת ניסוי המתבססת על ערכת לגו NXT וסביבת עבודה יעודית בתוכנת MATLAB Simulink על מנת להדגים את ביצועי הבקר.

אוניברסיטת בן-גוריון בנגב

הפקולטה למדעי ההנדסה

המחלקה להנדסת מכונות

ייצוב של מערכת רובוטית תת-ממונעת

על כדור

מאת:

אלון אוהב-ציון

מנחה: דר. אמיר שפירא

חיבור זה מהווה חלק מהדרישות

לקבלת התואר "מגיסטר" בהנדסה

חשוון תש"ע **דצמבר 2009**